

预约下单功能已经完成，此时我们需要让用户支付支付体检相关的费用，现在比较流行支付方式有：

1. 支付宝支付（我们使用支付宝）
2. 微信支付
3. 银联支付

1、对接支付宝的测试环境

1.1、使用自己支付宝扫码登录 <https://open.alipay.com/platform/home.htm>

1.2、第一次使用的需要填写一下资料，然后提交申请就可以使用了

<https://open.alipay.com/platform/entryGuidance.htm#/>

入驻身份：系统服务商ISV [切换入驻身份>](#)

填写信息

* 联系人姓名：

* 联系人手机：

* 联系人邮箱：

* 所在地：

详细地址：

业务简介：

您感兴趣的行业 ②：

您的线下推广资源范围 ②：

您业务覆盖地区 ②：

1.3、使用沙箱环境（支付宝的测试环境）



1.4、下载密钥生成工具

沙箱应用 i

o 信息配置

必看部分

| | |
|---|--|
| APPID i | 2016101200668686 |
| 支付宝网关 i | https://openapi.alipay.com/gateway.do |
| RSA2(SHA256)密钥(推荐) i | 设置 |

开发者要保证接口中使用的私钥与此处的公钥匹配，否则无法调用接口。可参考密钥的生成方法。沙箱支付宝公钥与线上不同，

1.5、密钥生成文档 <https://docs.open.alipay.com/291/105971>

第一步：生成 RSA 密钥

更新时间：2019-09-20

工具下载

支付宝为技术开发人员提供一键生成工具，便于开发者生成一对 RSA 密钥、公钥证书申请CSR文件（在线申请应用公钥证书需要）。

应用公钥（public key）需提供给支付宝账号管理者上传到支付宝开放平台；应用私钥（private key）由开发者自己保存，需填写到代码中供签名时使用。加密的过程为系统使用公钥（public key）进行加密，并将密文发送到解密者，解密者用私钥（private key）解密将密文解码为明文。

开发者要保证接口中使用的私钥与此处的公钥匹配，否则无法调用接口。开发者可通过下方链接下载对应的密钥生成工具：

[WINDOWS](#) (windows版本工具请不要安装在含有空格的目录路径下)

[MAC_OSX](#)

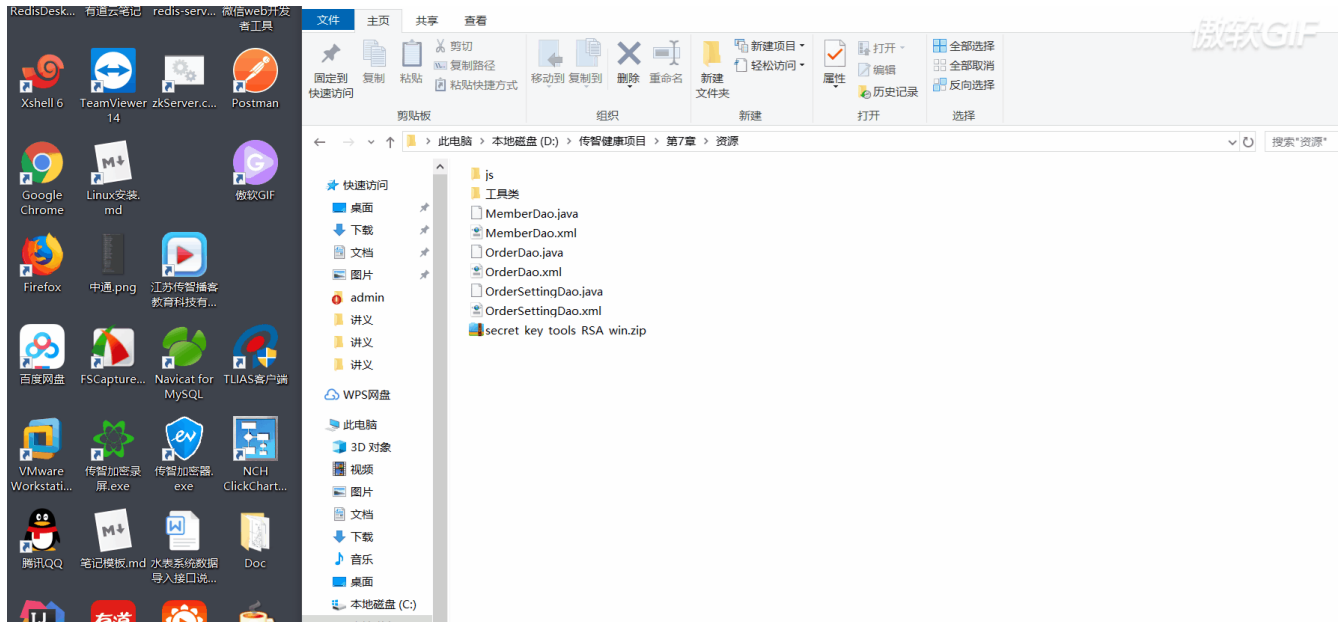
WINDOWS老版本下载地址： [WINDOWS](#) 此处我下载老版本

MAC OS老版本下载地址： [MAC_OSX](#)

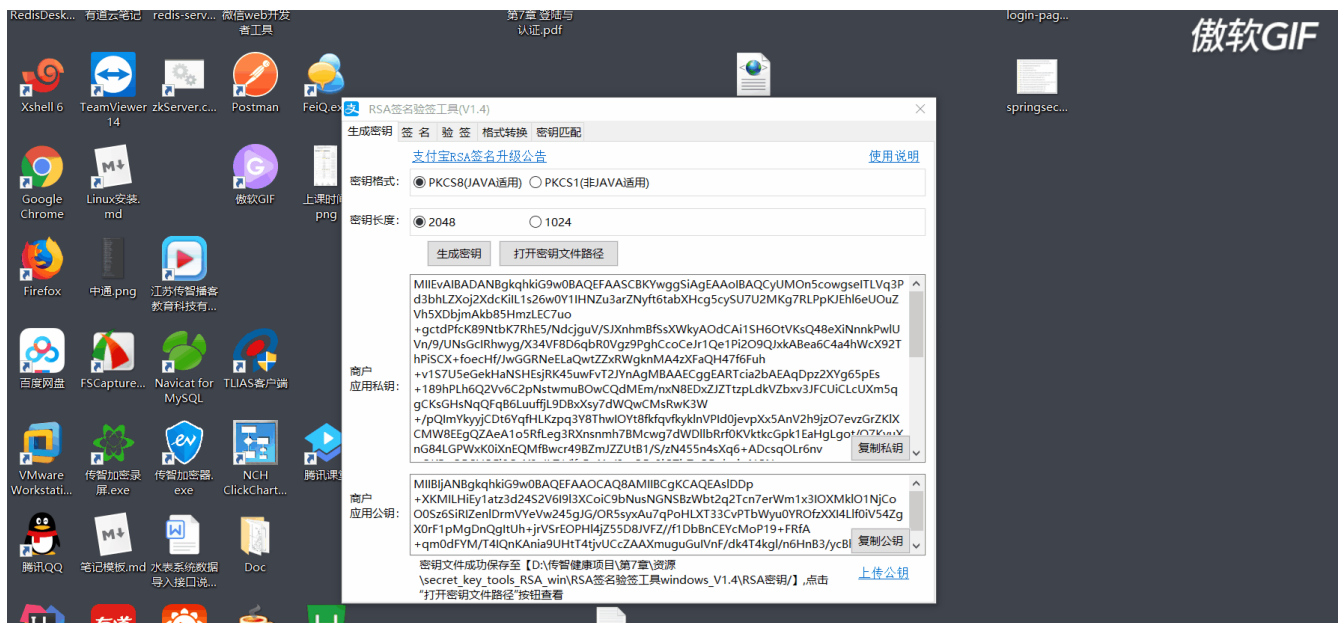
注意：

密钥和应用（APPID）一一对应，即开发者需要为名下的每个应用分别设置密钥，且不同应用的密钥不能混用。

1.6、解压密钥生成工具&生成密钥



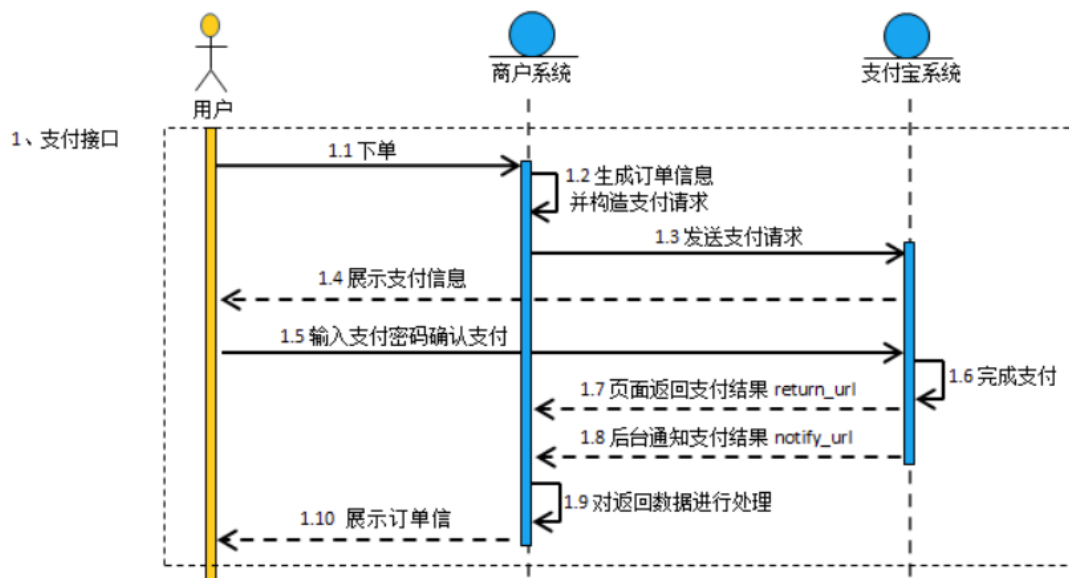
1.7、在上传生成的应用公钥到支付宝



2、支付宝配置完成之后开始写对接支付宝代码

2.1、了解支付流程 <https://docs.open.alipay.com/203/105285/>

2.2、调用流程



2.3、引入支付sdk

```

<!-- https://mvnrepository.com/artifact/com.alipay.sdk/alipay-sdk-java -->
<dependency>
    <groupId>com.alipay.sdk</groupId>
    <artifactId>alipay-sdk-java</artifactId>
    <version>4.7.12.ALL</version>
</dependency>
  
```

2.4、订单表添加支付状态字段

```

alter table t_order add pay_status int(11) default '0' comment '0 未支付 1 支付中 2 支付成功 3 支付失败'
  
```

2.5、修改下单成功ajax回调

原来下单成功直接跳转到支付页面，现在需要调用支付接口

```
axios.post("/order/submit.do",this.orderInfo).then((response) => {  
    if(response.data.flag){  
        //预约成功，跳转到预约成功页面  
        // window.location.href="orderSuccess.html?orderId=" +  
response.data.data.id;  
        //跳转到支付页面  
        window.location.href = "/pay/alipay.do?orderId=" +  
response.data.data.id;  
    }else{  
        //预约失败，提示预约失败信息  
        this.$message.error(response.data.message);  
    }  
});
```

2.6、Controller

```
import cn.hutool.core.date.DateUtil;
import com.alibaba.dubbo.config.annotation.Reference;
import com.alibaba.fastjson.JSONObject;
import com.alipay.api.AlipayApiException;
import com.alipay.api.AlipayClient;
import com.alipay.api.DefaultAlipayClient;
import com.alipay.api.internal.util.AlipaySignature;
import com.alipay.api.request.AlipayTradeWapPayRequest;
import com.itheima.service.OrderService;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseBody;
import org.springframework.web.bind.annotation.RestController;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.util.Date;
import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;

@RestController
@RequestMapping("/pay")
public class PayController {

    @Reference
    OrderService orderService;

    Logger log = LoggerFactory.getLogger(PayController.class);
    //支付宝服务器地址
    String SERVERURL = "https://openapi.alipaydev.com/gateway.do";
    //商户appid
    String APP_ID = "2016101300676878";
    //商户私钥

    String APP_PRIVATE_KEY =
```

```
"MIIIEvgIBADANBgkqhkiG9w0BAQEFAASCBAKgwggSkAgEAAoIBAQCcgWn1/l0gpxNvmWoAfRjG
lFmvjBa6o7r1M9kckzyfiE+CKkRKmHG06RBBs1MJnmpMMuaBfc0aHCZfsJxEEENyOJCgTlRtT
pztG/dM8t1PJJZqXG/oosCIhSj+Sk/zL7QL7a6Lsb3RcFk4Pp76f9Gn5y8NcROtTxvoUmEiEy7
TQtD8khbxP9jN3YQhHoWgshFRTdsuSMgx2rdeG9KuotYKgnub8I1PRFvAK7nZ2NsnT/D270QW
R+S07Lr4ewFWRzJOaIEJMW8meTmXbfr7lUbjepXSNP5PLXx18xwdOZAcNbrqP0gw+ZnYTq+x
6ieh1vs18tZTaCONHZ90+P3pTaYpAgMBAAECggEAB2fEOMSRaYMeU1CCCIXigH5yg/Dv/mL57
NlIM6bv3miXwoP3eUQfVKu6fjF05oydYP+jkor+Unb0heCYdEwEtPniNgOez9ZPbJS/Ld/sAt
u3peDIBHbZFpQpz+i6tHeBhZ4BNojLi7Pue6UnSvBI24hz1LfNjUNDj5qaAYvhxSQq4+HQDz
78Te52Dg+Q0iGEaSBAEJzbAfx2KAiRBRx3JZwvHv5Q9tLspBVBupd8CE2fOC6/QmYoCc1vtqk
sGsQjBaPek73N2GBmwp18fz2qY5QrXdzWUFhX0R1PBkQhjAUr0cuDHS0KNZ8zRjiEiK/W99yq
TKjNUqNVXj1VHaKBQKBgQDzy7V+xGrkWxE0aDqov/bNOPc5uJeX+RsnFmsYSujh4061GjFm/C
oNgdWGoqTV9jDLWzAIishi40NUUDXchFO0UtIk8p2zm2wVvozW6ntEXe3bcdJKPu0XM2EhGO
as0JJZ5yl30k2zYiUwcBOW+RU2fDZDOjhqZkIFqnW0oeG0ZwKBgQCkVxA8P1Y2PsPLURi1r8+K
uN9HfQXiViNtRq8ljUqxVIb75gKcxLLNvlehDKo8TSx4EPI7imc4qAQF1XF/KBEFsgk/Cj1rm
i5+Mtt8i43YlJfG4eZqtbIZ4DgZmcKsFMc5G9zTncxLD/1hKCXyOCkq9XBqbbJ0jicV4mehWs
As7wKBgFXrkD5qCOYL9jnmn+3DIb73+aFdHggQYfJnozy/gc3CZ4XKAmkyK7vw6wj1HzTM/Ct
fKtGH39GCGMOJNUJPXJSrr6YX315iQzhynIc8wZFSW43I4bg1+md5YXD3m4b08EhxAJslcCbt
SvbabwYq8r/uDM2qoiFVgqUuixnGkZC3AoGBAKJGvmdefq7t51S65xqDsoKe7wuebtMA9Q5Qv
95q2Jimi0vwQiAfMxhqngkPgoIupUTiJu+YSTju35oLHBmdrMfc/jhbm17IfqRq/8TJ4PU+NW
lT47z3UbivaXYOG0bnVZoU40erB/IPBfqMNHwr7KrY1Xqie5cQT5QSjpYMElvNAoGBANZHfKN
OzxTGM7g0AS6HauuVUjM09sYbA2n6m07nK6SlydT50nbptn9MpL8ddvMQiSGvivnBzE417P
GPimINA+kjZ8NKVvQwKa4vUEPFTLaANWvtiDmHsPVJm04YrBfEHwGjfmBJJ67NUR3GjiQ+ArY
rIfhQ0+aaWQ//k1LER7";
```

```
String CHARSET = "UTF-8";
```

```
//支付宝公钥
```

```
String ALIPAY_PUBLIC_KEY =
```

```
"MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAYCJ6Af6PEb+PXgc4qhxEs5We6YWC
yO+9T84Ib+WtwfpPkBsHOeVaAvL2Nz6xpNfMwTm+PC/+NXZJbHK90BsDBGW/QPENI/2qw4oWd
HIJBbBm09cGKrSUPfid0GWT24oTCvy8Tgiq5Vf3p+xWHaujBJCJqAJduvOPPEb70VP5uyx0I
n57q7pbMYU6NUMIZCvrzaBcBLPjjav41Q8CDkeDACZhjER7Cw/M+Blkk9dfrwBLoi1IhoIaqX
s+KjHnqP47wrxoFPRX0hONVvFFnrMME7i0MM6X22rI1a1uhvcieGPDwFS/pvuRAM6na0uJ4T0
uvmSeumSANRW51Awk0QipQIDAQAB";
```

```
/**
```

```
 * 构造支付请求
```

```
 * @param httpRequest
```

```
 * @param httpResponse
```

```
 * @param orderId
```

```
 * @throws ServletException
```

```
 * @throws IOException
```

```

    */
    @RequestMapping("/alipay")
    public void alipay(HttpServletRequest httpRequest,
        HttpServletResponse httpResponse,Integer orderId)
        throws ServletException, IOException {
        AlipayClient alipayClient =
            new DefaultAlipayClient(SERVERURL,
                APP_ID, APP_PRIVATE_KEY,
                "json",
                CHARSET,
                ALIPAY_PUBLIC_KEY, "RSA2"); //获得初始化的
AlipayClient

        AlipayTradeWapPayRequest alipayRequest = new
AlipayTradeWapPayRequest();//创建API对应的request

        alipayRequest.setReturnUrl("http://5muhnf.natappfree.cc/pages/paySuccess.
html");//支付成功之后跳转的地址(由于我们没有域名,我们使用natapp内网穿透工具)

        alipayRequest.setNotifyUrl("http://5muhnf.natappfree.cc/pay/alipayNotifyN
otice.do");//支付成功通知地址

        Map orderInfo = orderService.findById(orderId);

        JSONObject json = new JSONObject();
        json.put("out_trade_no",outTradeNo(orderId));//商户订单号, 需要保证
不重复(支付回调会带上这个参数,我们可以用来查询对应的订单修改状态)
        json.put("total_amount",1000);//订单金额
        json.put("subject",orderInfo.get("setmeal"));//订单标题
        alipayRequest.setBizContent(json.toJSONString());

        String form="";
        try {
            form = alipayClient.pageExecute(alipayRequest).getBody(); //
调用SDK生成表单
        } catch (AlipayApiException e) {
            e.printStackTrace();
        }
        httpResponse.setContentType("text/html;charset=" + CHARSET);

        httpResponse.getWriter().write(form);//直接将完整的表单html输出到页

```


面

```
        httpResponse.getWriter().flush();
        httpResponse.getWriter().close();
    }

    /**
     * 生成商户订单号（年月日时分秒 + 订单号）
     * @param orderId
     * @return
     */
    private String outTradeNo(Integer orderId){
        String dateStr = DateUtil.format(new Date(), "yyyyMMddHHmmss");
        return dateStr + orderId;
    }

    /**
     * 支付宝通知我们支付状态，在这个接口里面可以修改订单支付的状态
     * @param request
     * @param response
     * @return
     * @throws Exception
     */
    @RequestMapping(value = "/alipayNotify")
    @ResponseBody
    public String alipayNotify(HttpServletRequest request,
        HttpServletRequest response) throws Exception {

        log.info("支付成功，进入异步通知接口...");

        //获取支付宝POST过来反馈信息
        Map<String,String> params = new HashMap<String,String>();
        Map<String,String[]> requestParams = request.getParameterMap();
        for (Iterator<String> iter = requestParams.keySet().iterator();
            iter.hasNext();) {
            String name = (String) iter.next();
            String[] values = (String[]) requestParams.get(name);
            String valueStr = "";
            for (int i = 0; i < values.length; i++) {
                valueStr = (i == values.length - 1) ? valueStr +
values[i]

                : valueStr + values[i] + ",";
            }
        }
    }
}
```

```

    }
    //乱码解决，这段代码在出现乱码时使用
    valueStr = new String(valueStr.getBytes("ISO-8859-1"), "utf-8");

    params.put(name, valueStr);
}
boolean signVerified = AlipaySignature.rsaCheckV1(params,
ALIPAY_PUBLIC_KEY, "utf-8", "RSA2"); //调用SDK验证签名
//—请在这里编写您的程序（以下代码仅作参考）—
/* 实际验证过程建议商户务必添加以下校验：
1、需要验证该通知数据中的out_trade_no是否为商户系统中创建的订单号，
2、判断total_amount是否确实为该订单的实际金额（即商户订单创建时的金
额），
3、校验通知中的seller_id（或者seller_email）是否为out_trade_no这笔单
据的对应的操作方（有的时候，一个商户可能有多个seller_id/seller_email）
4、验证app_id是否为该商户本身。
*/
if(signVerified) { //验证成功
    //商户订单号
    String out_trade_no = new
String(request.getParameter("out_trade_no").getBytes("ISO-8859-1"), "UTF-8");

    //支付宝交易号
    String trade_no = new
String(request.getParameter("trade_no").getBytes("ISO-8859-1"), "UTF-8");

    //交易状态
    String trade_status = new
String(request.getParameter("trade_status").getBytes("ISO-8859-1"), "UTF-8");

    //付款金额
    String total_amount = new
String(request.getParameter("total_amount").getBytes("ISO-8859-1"), "UTF-8");

    if(trade_status.equals("TRADE_FINISHED")){
        //判断该笔订单是否在商户网站中已经做过处理

        //如果没有做过处理，根据订单号（out_trade_no）在商户网站的订单

```

系统中查到该笔订单的详细，并执行商户的业务程序

//如果有做过处理，不执行商户的业务程序

//注意： 尚自习的订单没有退款功能，这个条件判断是进不来的，所以此处不必写代码

//退款日期超过可退款期限后（如三个月可退款），支付宝系统发送该交易状态通知

```
}else if (trade_status.equals("TRADE_SUCCESS")){
```

//判断该笔订单是否在商户网站中已经做过处理

//如果没有做过处理，根据订单号（out_trade_no）在商户网站的订单系统中查到该笔订单的详细，并执行商户的业务程序

//如果有做过处理，不执行商户的业务程序

//注意：

//付款完成后，支付宝系统发送该交易状态通知

// 修改叮当状态，改为 支付成功，已付款；同时新增支付流水

```
}
```

```
log.info("支付成功...");
```

```
}else { //验证失败
```

```
log.info("支付，验签失败...");
```

```
}
```

```
return "success";
```

```
}
```

```
}
```