# 法一 个人直接APP申请（推荐）

1. 打开支付宝APP搜索“蚂蚁金服商家平台”点击进入



1. 点击右上方开通支付图标



1. 点击立即签约



1. 填写信息 门头照自行解决，如去美团外卖中商家信息找别的店铺照片



# 法二 商家代签约（需2个支付宝账号）

1. 登录支付宝进入商家签约管理

<https://openhome.alipay.com/isv/settling/queryMerchantSignProductList.htm#/>



1. 输入要开通的支付宝账号（不能是自己的）



1. 选择当面付



1. 填写信息同上



此处仅作JAVA中开发简单说明，详见官方文档和代码

# 开发前期准备

1. 签约成功后登录蚂蚁金服开放平台 点击右上角头像 进入秘钥管理

<https://open.alipay.com/platform/manageHome.htm>

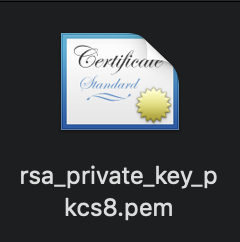


1. 根据提示配置私钥和公钥，私钥根据提示下载支付宝RAS密钥生成器即可

<https://docs.open.alipay.com/291/105971>



查看私钥文件 后期代码中将用到



注意代码中要用的2个配置已圈出，AppID为应用ID，公钥为支付宝公钥，非应用公钥，还有一个为上图中的私钥内容，用编辑工具打开私钥内容复制即可



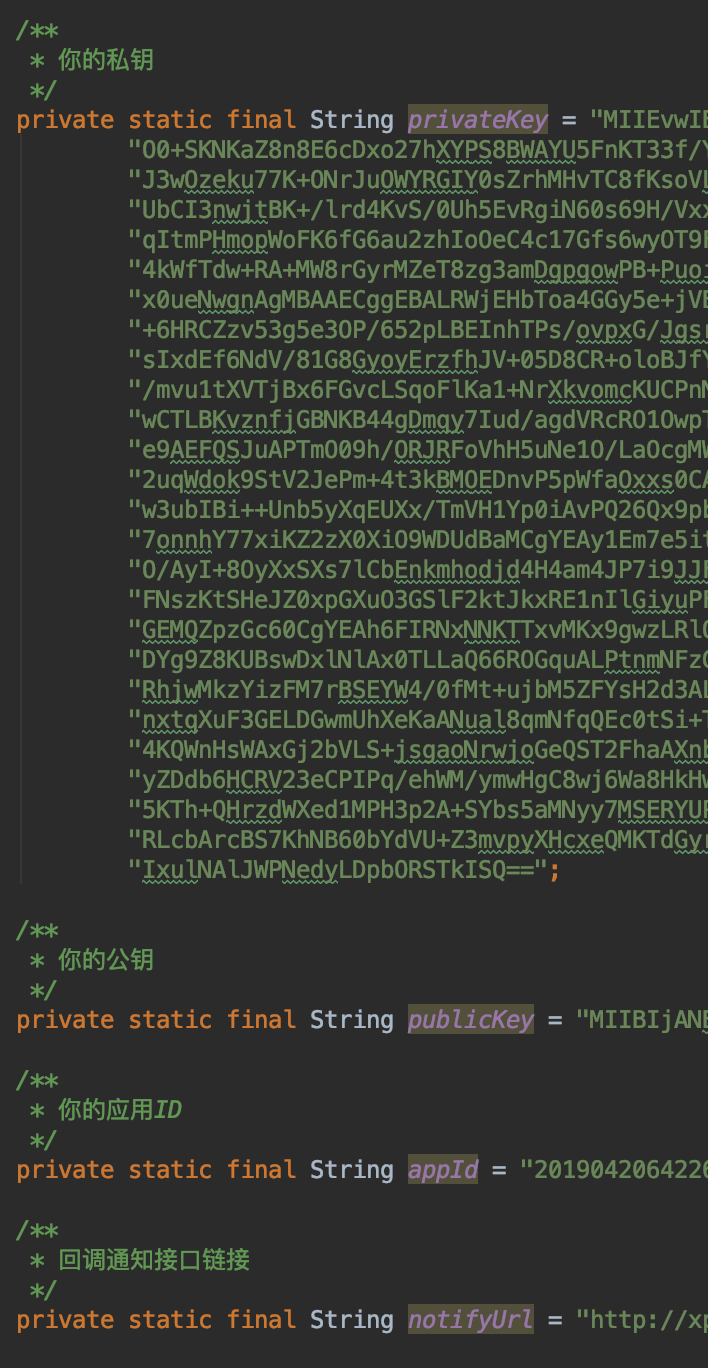
# 开发简介

1. 引入maven依赖

<dependency>  
 <groupId>com.alipay.sdk</groupId>  
 <artifactId>alipay-sdk-java</artifactId>  
 <version>3.7.26.ALL</version>  
</dependency>

1. 开发即可

详见AlipayController，填入你的配置 注意修改你的服务器回调链接



先自己生成订单号调用生成二维码，页面轮询再根据订单号查询的支付状态，官方也提供了notify\_url通知链接以供回调

/\*\*  
 \* 生成二维码  
 \* @param pay  
 \* @param request  
 \* @return  
 \* @throws AlipayApiException  
 \*/  
@RequestMapping(value = "/alipay/precreate",method = RequestMethod.POST)  
@ResponseBody  
public Result<Object> getPayState(@ModelAttribute Pay pay,) throws AlipayApiException {  
  
 AlipayClient alipayClient = new DefaultAlipayClient("https://openapi.alipay.com/gateway.do",  
 appId,privateKey,"json","GBK",publicKey,"RSA2");  
 AlipayTradePrecreateRequest r = new AlipayTradePrecreateRequest();  
 r.setBizContent("{" +

"\"notify\_url\":\""+notifyUrl+"\"," +  
 "\"out\_trade\_no\":\"你的订单号id\"," +  
 "\"total\_amount\":填金额数字如：0.01," +  
 "\"subject\":\"支付标题如：XPay-向作者Exrick捐赠\"" +  
 " }");  
 AlipayTradePrecreateResponse response = alipayClient.execute(r);  
 if(!response.isSuccess()){  
 return new ResultUtil<Object>().setErrorMsg("调用支付宝接口生成二维码失败，请向作者反馈");  
 }  
 Map<String, Object> result = new HashMap<>(16);  
 result.put("id", pay.getId());  
 result.put("qrCode", response.getQrCode());  
 return new ResultUtil<Object>().setData(result);  
}

/\*\*

\* 支付宝通知回调

\* @return

\*/

@RequestMapping(value = "/alipay/notify",method = RequestMethod.GET)

@ResponseBody

public String notify(@RequestParam String out\_trade\_no,

@RequestParam String trade\_status) {

if("TRADE\_SUCCESS".equals(trade\_status)){

// 支付成功

}

return "success";

}  
  
/\*\*  
 \* 查询支付结果  
 \* @param outTradeNo  
 \* @return  
 \* @throws AlipayApiException  
 \*/  
@RequestMapping(value = "/alipay/query/{outTradeNo}",method = RequestMethod.GET)  
@ResponseBody  
public Result<Object> queryPayState(@PathVariable String outTradeNo) throws AlipayApiException {  
  
 AlipayClient alipayClient = new DefaultAlipayClient("https://openapi.alipay.com/gateway.do",  
 appId,privateKey,"json","GBK",publicKey,"RSA2");  
 AlipayTradeQueryRequest request = new AlipayTradeQueryRequest();  
 request.setBizContent("{" +  
 "\"out\_trade\_no\":\""+outTradeNo+"\"" +  
 " }");  
 AlipayTradeQueryResponse response = alipayClient.execute(request);  
 if(response!=null&&response.isSuccess()&&"TRADE\_SUCCESS".equals(response.getTradeStatus())){  
 // 支付成功

return new ResultUtil<Object>().setData(1);  
 }else{

// 等待支付  
 return new ResultUtil<Object>().setData(0);  
 }  
}

本demo通过前端轮询查询订单支付状态+官方通知回调+每日23:55再次检查订单，你也可以更加完善避免漏单，如增加定时任务、页面显示询问用户是否支付等