

支付宝支付

1. 支付宝开放平台登录

使用已有的支付宝账号即可

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

2. 关于沙箱环境（开发模拟环境）

沙箱环境使用说明

更新时间：2017-09-19

蚂蚁沙箱环境(Beta)是协助开发者进行接口功能开发及主要功能联调的辅助环境。沙箱环境模拟了开放平台部分产品的主要功能和主要逻辑（当前沙箱支持产品请参考“沙箱支持产品列表”）。

在开发者应用上线审核前，开发者可以根据自身需求，先在沙箱环境中了解、组合和调试各种开放接口，进行开发调通工作，从而帮助开发者在应用上线审核完成后，能更快速、更顺利的进行线上调试和验收工作。

如何使用和配置沙箱环境请参考《沙箱环境使用说明》。

注意：

- 由于沙箱为模拟环境，在沙箱完成接口开发及主要功能调试后，请务必在蚂蚁正式环境进行完整的功能验收测试。所有返回码及业务逻辑以正式环境为准。
- 为保证沙箱稳定，沙箱环境测试数据会进行定期数据清理。Beta测试阶段每周日中午12点至每周一中午12点为维护时间。在此时间内沙箱环境部分功能可能会不可用，敬请谅解。
- 请勿在沙箱进行压力测试，以免触发相应的限流措施，导致无法正常使用沙箱环境。
- 沙箱支持的各个开放产品，沙箱使用的特别说明请参考各产品的快速接入文档或技术接入文档章节。

<https://docs.open.alipay.com/200/105311>

登录后，在页面顶部可以选择进入沙箱环境设置页面



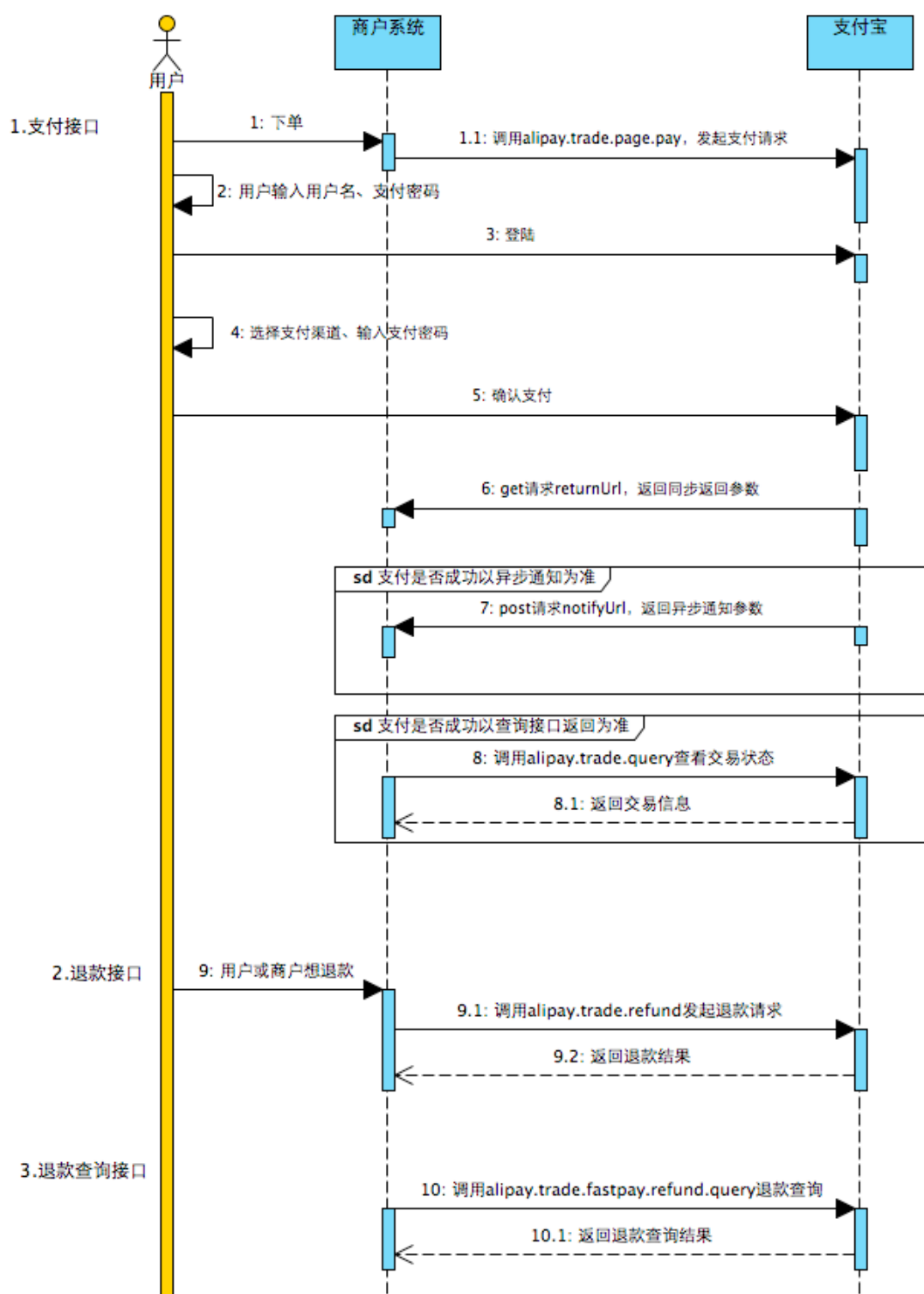
3. 支付宝开发者文档

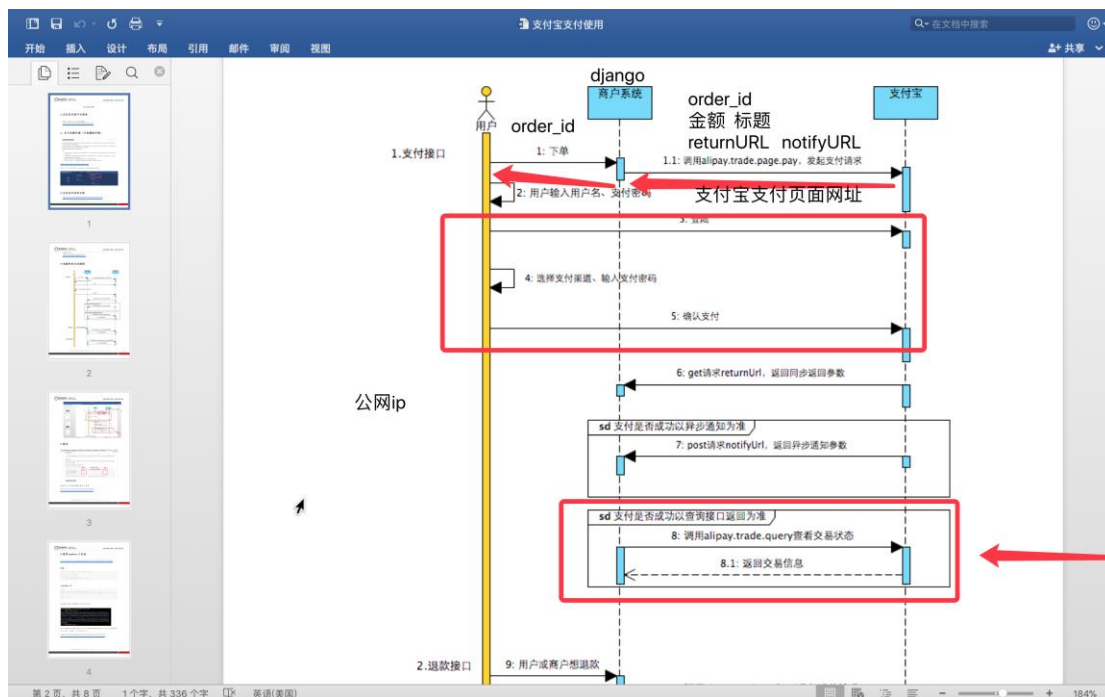
<https://openhome.alipay.com/developmentDocument.htm>

电脑网站支付

<https://docs.open.alipay.com/270>

4. 电脑网站支付流程





5. 签名

则待签名字符串为:

```
app_id=2014072300007148&biz_content={"button":[{"actionParam":"ZFB_HFCZ","actionType":"out","name":"话费充值"}]}
```

3.调用签名函数

使用各自语言对应的SHA256WithRSA(对应sign_type为RSA2)或SHA1WithRSA(对应sign_type为RSA)签名函数利用商户私钥对待签名字符串进行签名, 并进行Base64编码。

4.把生成的签名赋值给sign参数, 拼接到请求参数中。

可以参考[开放平台SDK](https://docs.open.alipay.com/291/106118)源码中AlipaySignature.rsaSign方法。

django

数据进行签名计算

私钥 → 公钥

alipay

```
/**
 * @param content 加签内容
 * @param privateKey 加签私钥
 * @param charset 加签字符集
 * @param charset 签名方法
 */
String AlipaySignature.rsaSign(String content, String privateKey, String charset,String signType)
```

工具演示签名过程:

发送给支付的请求都需要进行签名

<https://docs.open.alipay.com/291/106118>

6. 使用 python 工具包

<https://github.com/fzlee/alipay/blob/master/README.zh-hans.md>

安装

```
# 从 1.3.0 升级上来的用户，请先卸载 pycrypto:
pip uninstall pycrypto
# 安装 python-alipay-sdk
pip install python-alipay-sdk --upgrade
```

生成密钥文件

```
openssl
OpenSSL> genrsa -out app_private_key.pem 2048 # 私钥
OpenSSL> rsa -in app_private_key.pem -pubout -out app_public_key.pem # 导出公钥
OpenSSL> exit
```

cat app_public_key.pem 查看公钥的内容

```
delron@D-MBP:~$ cat app_public_key.pem
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAYoXC1zh3JKLB0ypAf4hr
tMqRn10kXsUI3W8j8z6vX0m+hG7cq1MGarkb1HdmxX7dQLcbFdoq3BjXGzxKH+AP
o5PUSbLXYlvYmE79bK36jb8+XZaIpu3odWzwpztREXNHLHltJhYRix6rrKbB+PiR
pRo/zZVecebNhkLFlyIz5DsC2K060g/8Ck+DRancbgkgYU8iv1RzNVuG8vVASZC
RXyAlJMLf50WIfnpLIyD3LzK0PrWN0yrQsYUmGC57DbfyTEl87YrqWre9nMHUzUZ
+V+e+sV13JViJi9bYDzFdVaufktatPfVDG1yNA713G1UeWoZjYcYu9FwqIc1EQqE
CQIDAQAB
-----END PUBLIC KEY-----
```

将-----BEGIN PUBLIC KEY-----和-----END PUBLIC KEY-----中间的内容保存在支付宝的用户配置中（沙箱或者正式）

<https://openhome.alipay.com/platform/appDaily.htm?tab=info>

沙箱应用 i

信息配置

必看部分

APPID i	2016081600258081
支付宝网关 i	https://openapi.alipaydev.com/gateway.do
RSA2(SHA256)密钥(推荐) i	查看应用公钥 查看支付宝公钥

应用公钥(SHA256withRsa) ×

② 应用调用接口时的加签格式同时支持公钥和公钥证书，开发者可根据自己的实际情况选择使用，如果同时设置了应用公钥和公钥证书，则两种方式的公钥都可生效 使用SHA256withRsa，支付宝会用SHA256withRsa算法进行接口调用时的密钥解密（不限密钥长度）

应用公钥

[修改](#)

MIIBjANBgqhkiG9w0BAQEFAAQCAQ8AMIIBCgKCAQEAYoXC1zh3JkLBOypAf4hr tMqRn1OkXsUI3W8j8z6vX0m+hG7c qIMGarkb1HdmxX7dQLcbFdoq3BjXGzxKH+AP o5PUSbIXYivYmE79bK36jb8+XZalpu3odWzwpztREXNHLHltJhYRix6rrKb B+PIR pRo/zZVecebNhkLFYilz5DsC2K060g/8Ck+DRancbgkgYU8iv1RzNVuG8vVASZC RYyAJJMLf50WfnpLlyD3LzK0Pr WNOyrQsYUmGC57DbfyTEI87YrqWre9nMHUzUz +V+e+sV13JViJi9bYDzFdVaufktatPFDVG1yNA713G1UeWoZjYcYu9Fwq lc1EQqE CQIDAQAB

应用公钥证书 （仅支持X509 V3格式的证书文件，扩展名为.crt）

[上传](#)

暂无

应用公钥(SHA256withRsa) ×

② 使用SHA256withRsa，支付宝会用SHA256withRsa算法进行接口调用时的验签（不限密钥长度）。[查看密钥生成方法](#)

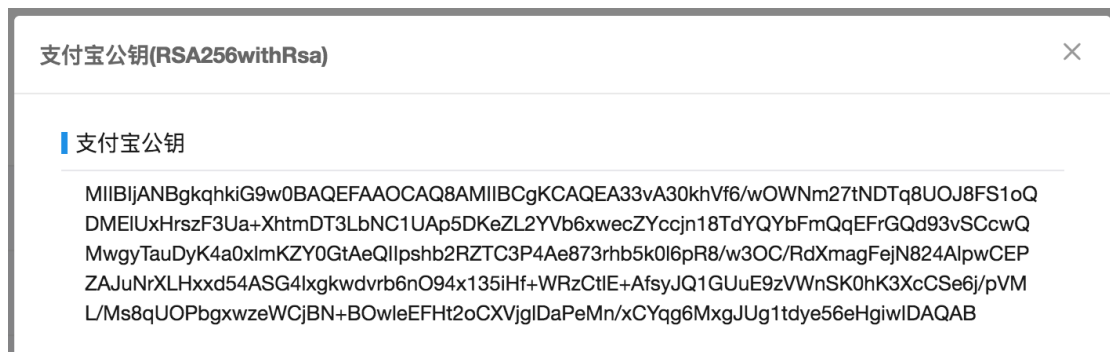
MIIBjANBgqhkiG9w0BAQEFAAQCAQ8AMIIBCgKCAQEAYoXC1zh3JkLBOypAf4hr tMqRn1OkXsUI3W8j8z6vX0m+hG7c qIMGarkb1HdmxX7dQLcbFdoq3BjXGzxKH+AP o5PUSbIXYivYmE79bK36jb8+XZalpu3odWzwpztREXNHLHltJhYRix6rrKbB+PIR pRo/zZVecebNhkLFYilz5DsC2K060g/8Ck+DRancbgkgYU8iv1RzNVuG8vVASZC RYyAJJMLf50WfnpLlyD3LzK0PrWNOyrQsYUmGC57DbfyTEI87YrqWre9nMHUzUz +V+e+sV13JViJi9bYDzFdVaufktatPFDVG1yNA713G1UeWoZjYcYu9Fwqlc1EQqE CQIDAQAB

[验证公钥正确性>>](#)

保存

下载支付宝的公钥文件

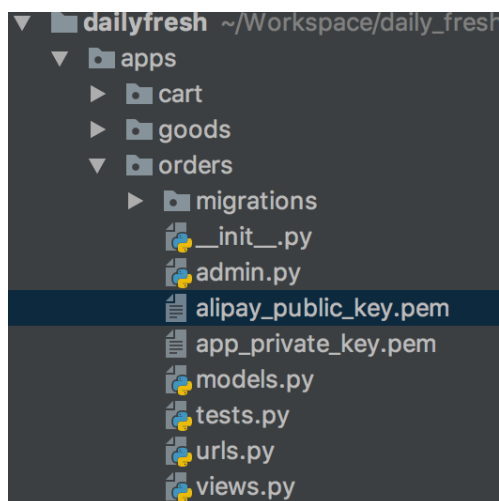
APPID ⓘ	2016081600258081
支付宝网关 ⓘ	https://openapi.alipaydev.com/gateway.do
RSA2(SHA256)密钥(推荐) ⓘ	查看应用公钥 查看支付宝公钥
RSA(SHA1)密钥 ⓘ	查看应用公钥 查看支付宝公钥



将公钥的内容复制保存到一个文本文件中(alipay_public_key.pem)，注意需要在文本的首尾添加标记位(-----BEGIN PUBLIC KEY-----和-----END PUBLIC KEY-----)，形如：

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA33vA30khVf6/wOWNm27tNDTq8UOJ8FS1oQ
-----END PUBLIC KEY-----
```

将刚刚生成的私钥 app_private_key.pem 和支付宝公钥 alipay_public_key.pem 放到我们的项目目录中



使用支付宝 python 包的初始化

```

alipay = AliPay(
    appid=settings.ALIPAY_APPID,
    app_notify_url=None, # 默认回调url
    app_private_key_path=os.path.join(settings.BASE_DIR, "apps/orders/app_private_key.pem"),
    alipay_public_key_path=os.path.join(settings.BASE_DIR, "apps/orders/alipay_public_key.pem"),
    sign_type="RSA2", # RSA 或者 RSA2
    debug=True # 默认False
)

```

7. 支付接口

<https://docs.open.alipay.com/270/alipay.trade.page.pay/>

```

def order_pay(request):
    '''订单支付'''
    # 登录状态检查
    if not request.session.has_key('islogin'):
        return JsonResponse({'res':0, 'errmsg':'用户未登录'})

    # 获取订单编号
    order_id = request.POST.get('order_id')

    # 数据校验
    if not order_id:
        return JsonResponse({'res':1, 'errmsg':'缺少订单标号'})

    # 判断订单信息是否正确
    try:
        order = OrderInfo.objects.get(order_id=order_id,
                                       pay_method=3,
                                       status=1)
    except OrderInfo.DoesNotExist:
        return JsonResponse({'res':2, 'errmsg':'订单信息有误'})

    # 构造支付宝链接地址
    alipay = AliPay(
        appid="2016090800464054",
        app_notify_url=None, # 默认回调url
        app_private_key_path=os.path.join(settings.BASE_DIR, 'df_order/app_private_key.pem'),
        alipay_public_key_path=os.path.join(settings.BASE_DIR, 'df_order/alipay_public_key.pem'),
        sign_type="RSA2", # RSA 或者 RSA2
        debug=True # 默认False
    )

```

```

# 电脑网站支付, 需要跳转到https://openapi.alipay.com/gateway.do? + order_string
total_pay = order.total_price + order.transit_price
order_string = alipay.api_alipay_trade_page_pay(
    out_trade_no=order_id,
    total_amount=str(total_pay),
    subject='天天生鲜%s'%order_id,
    return_url=None,
    notify_url=None, # 可选, 不填则使用默认notify url
)

alipay_url = settings.ALIPAY_URL + '?' + order_string
return JsonResponse({'res':3, 'message':'OK', 'pay_url':alipay_url})

```

8. 获取支付结果接口

https://docs.open.alipay.com/api_1/alipay.trade.query

```

def check_pay(request):
    '''支付结果检查'''
    # 登录状态检查
    if not request.session.has_key('islogin'):
        return JsonResponse({'res': 0, 'errmsg': '用户未登录'})
    #
    order_id = request.POST.get('order_id')

    if not order_id:
        return JsonResponse({'res': 2, 'errmsg': '缺少订单编号'})

    passport_id = request.session.get('passport_id')
    try:
        order = OrderInfo.objects.get(order_id=order_id,
                                       passport_id=passport_id,
                                       status=1)
    except OrderInfo.DoesNotExist:
        return JsonResponse({'res': 3, 'errmsg': '订单信息错误'})

    alipay = AliPay(
        appid='2016090800464054',
        app_notify_url=None, # 默认回调url
        app_private_key_path=os.path.join(settings.BASE_DIR, 'df_order/app_private_key.pem'),
        alipay_public_key_path=os.path.join(settings.BASE_DIR, 'df_order/alipay_public_key.pem'),
        # 支付宝的公钥，验证支付宝回传消息使用，不是你自己的公钥，
        sign_type='RSA2', # RSA 或者 RSA2
        debug=True # 默认False
    )

```

```

while True:
    # 获取支付结果
    result = alipay.api_alipay_trade_query(order_id)

    code = result['code']

    if code == '10000':
        # 支付宝支付订单生成成功
        trade_status = result.get('trade_status')
        if trade_status is None:
            time.sleep(5)
            continue

        if trade_status == 'TRADE_SUCCESS':
            # 更新订单的支付状态
            order.status = 2
            # 保存支付交易编号
            order.trade_id = result['trade_no']
            order.save()
            return JsonResponse({'res': 4, 'message': '支付成功'})
        elif trade_status == 'WAIT_BUYER_PAY':
            # 等待用户支付
            time.sleep(5)
            continue
        else:
            # 支付失败
            return JsonResponse({'res': 5, 'errmsg': '支付失败'})

    elif code == '40004':
        # 继续查询
        time.sleep(10)
        continue
    else:
        # 支付失败
        return JsonResponse({'res': 5, 'errmsg': '支付失败'})

```