

第三方支付-支付宝接入相关

第三方支付-支付宝接入相关

- 1.使用openssl库生成一对密钥（公钥和私钥）备用
- 2.支付宝创建应用
- 3.给应用添加功能
4. 给“电脑网站支付”功能接口加签
- 5.沙箱环境开发
(<https://docs.open.alipay.com/200/105311#s0>):
- 6.后端代码实现(Django):

1.使用openssl库生成一对密钥（公钥和私钥）备用

- 1.1 安装python-alipay-sdk

```
tarena@tedu:~$sudo pip3 install python-alipay-sdk --upgrade
```

- 1.2 生成密钥文件app_private_key.pem 和 app_public_key.pem

```
tarena@tedu:~$openssl
OpenSSL> genrsa -out app_private_key.pem 2048
#私钥
OpenSSL> rsa -in app_private_key.pem -pubout
-out app_public_key.pem #导出公
钥
OpenSSL> exit
tarena@tedu:~$ls
app_private_key.pem app_public_key.pem
```

2.支付宝创建应用

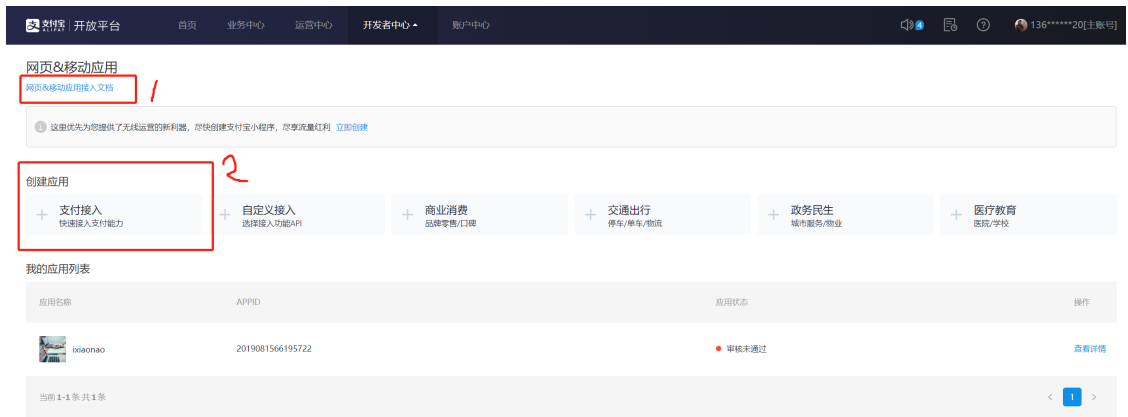
- 2.1 点击链接，进入支付宝开放平台(<https://openhome.alipay.com/platform/home.htm>)
- 2.2 使用支付宝账号登录（扫码或者账号密码登录）



- 2.3 点击“开发者中心”，选择“网页移动应用”



- 2.4 选择“支付接入”功能，创建一个应用。



- 2.5 应用设置, 填写必填项, 应用类型选择“网页应用”, 点击“确认创建”。

* 应用名称:
不超过32个字符, [查看命名规范](#)

* 应用图标:  [重新上传](#)  [示意图](#)
请上传应用高清图, 支持.jpg .jpeg .png格式, 建议320*320像素, 小于3M

应用类型: ☒ 网页应用 ☐ 移动应用

网址url:



应用简介:

[确认创建](#)

3.给应用添加功能

- 3.1 点击应用的“查看详情”

我的应用列表

应用名称	APPID	应用状态	操作
 达达商城edu	2019102968748710	开发中	查看详情
 xiaonao	2019081566195722	审核未通过	查看详情

当前 1-2 条 共 2 条

- 3.2 在功能列表中添加“电脑网站支付”功能(默认没有)

功能列表

添加功能

您已经添加4项功能

<input type="checkbox"/>	功能名称	功能介绍	是否需要商 户签约	状态	操作
<input type="checkbox"/>	快捷手机wap支付	商户在网页中调用支付宝提供的网页支付接口，通过网页跳转到支付宝中完成支付。	需签约	未签约	删除
<input type="checkbox"/>	APP支付	商户APP集成支付宝提供的SDK，通过SDK跳转到支付宝中完成支付，支付完后跳回商户页面。	需签约	未签约	删除
<input type="checkbox"/>	电脑网站支付	用户通过支付宝PC网站收银台完成支付，交易款项即时给到商户支付宝账户。	需签约	未签约	删除
<input type="checkbox"/>	当面付收单	买家通过支付宝钱包付款码、扫码的方式给商户付款，完成交易，商户快速收银，无需找零。	需签约	未签约	删除

确认已选功能

以下功能使用前需要完成签约：

- 电脑网站支付

确定

取消

4. 给“电脑网站支付”功能接口加签

- 4.1添加“电脑网站支付功能”后，进行开发设置，选择“接口加签方式”，点击“设置”

开发设置

①

使用支付宝的部分功能前，需要先设置应用环境，[查看如何使用](#)；如果需要升级RSA1到RSA2，[请参考操作指南](#)。

接口加签方式：

暂无

设置

IP白名单：

未设置

设置

支付宝网关：

https://openapi.alipay.com/gateway.do

应用网关：

暂无

设置

授权回调地址：

暂无

设置

将1.2中公钥app_public_key.pem的内容复制进去，保存设置，生成支付宝公钥，并保存在文件apipay_public_key.pem中。

加签管理

加签内容配置

加签配置完成

×

应用公钥:

[更换应用公钥](#)
[复制公钥](#)

MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAzPurRYGGjl+9OlnGxldS uh0U86QVJ7pEa7Twjd
vaxxL3pZJUtdcu/3cFhsbu9HUykdC3K9EyYs76CZOnKql9 iPwOug7x5xnneoN1mtDo2SfLZMsMtp3Xkzryu
Q9Sdx2BxXFx9QfrMQDOR6tGN Gum M2E46SIzb40116kbIXJ9HDLiM8Ra2kTGCW2+heFBgSJ0jh5al5stT4C
eu4E1oKX+ LmBIU+21Cqi1wedokedl26KrS87rFR55hopd+3zOkSxgNR3iVTIWr5jII3hVDDqx c2kyJt24GSL
MAxUWEXIJABYVs5SEHhLOYwWMAgWLDc57g8eFPi1LB6VTJ6xn+u8L LQIDAQAB

支付宝公钥:

[复制公钥](#)

MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAhMhz50Xkl6ISJhq6JNxHhkltr/hke2UrYeioROsR
n9zbErfKmM82WMqgurW7UaC1BUObD4HmFacHkXLW7+jqH2x7mENiOtSbvdC/5MoggFCLegOO0ehxo
blddeeeDuh1ar0hFzsnZBxezxtKVPbEuQYQofoifPLQFqy4AXmgjPwV49fHbPq4cUXUyuknN/HAvJxg9Z6R
M1AMKZt9LwfbIsqYB9iqAT+yjYH5J80a+i+BmidpXCK9mg2sYFNC8jHjj+36Q2OiHUQMxKbmLWOTQeXm
BDp+LFnsjc9K/ojBWokpDG+eyt0w2UNA20HJjynskZiEEGaLLpo9A+9o6JYQIDAQAB

[查看接入文档](#)

4.2 同意《支付宝业务协作协议》后点击“提交审核”

<input checked="" type="checkbox"/>	电脑网站支付	用户通过支付宝PC网站收银台完成支付，交易款项即时给到商户支付宝账户。	需签约	未签约	删除
<input type="checkbox"/>	当面付收单	买家通过支付宝钱包付款码、扫码的方式给商户付款，完成交易，商户快捷收银，无需找零。	需签约	未签约	删除

☒ 已阅读并同意 《支付宝业务协作协议》，《花呗分期产品推广协作协议》，《口碑业务协作协议》

概览

1 审核未通过，重新提交

2 等待审核结果

3 应用上线

1 开发中 请先完成添加功能、进行开发配置，提交审核通过后可正式上线

提交审核

删除

说明:

以上步骤为商家接入支付宝需要在开发平台做的操作,填写信息后提交审核通过后方可进行线上交易。

开发测试阶段可以使用沙箱环境进行开发

后续转为线上模式时只需要修改APPID即可代码不用动。

电脑支付 文档 <https://docs.open.alipay.com/270/105898/>

5.沙箱环境开发(<https://docs.open.alipay.com/200/105311#s0>):

- 5.1 登录到沙箱环境



- 5.2 关于沙箱：由支付宝官方虚拟的商家账户，开发者可以用来进行测试.需要先给沙箱配置公钥。需要记录 APPID和支付宝网关.



将应用公钥添加完毕后就就可以看到支付宝公钥,将支付宝公钥复制出来放到项目文件夹中

加签管理

加签内容配置

2 加签内容完成

×

应用公钥:

更换应用公钥 复制公钥

MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAqofrqM6B4x85W05so0cF B2Aq51BQ8WeOZyLUIXG4Ce/Oy+75euglOEbMvgzS0wvhnpiFtdgsTIOpCMcgbCoW 1ftUfnI51ynG1Rk51IMeydOBTWU0ze4tmnlvKtKXu9LJ6INZ8yyI2vFriHVZ/mnb eR5DW2bGx8/rkEqp+Fajxjag4rWLP7frnjQBrPpw8wxCS9ZECqnLJOJCxEgNnbhi qPKb0Gr0EuT5gOC7ipZNYu8yvQGLI33VIZBVHSGZ82uYP0BGGsUPh2lky5SLRZfv eDAItBTEpkE9wl33k8y/LQyPwD8pISJfsHXMIVa58fAH2IFvRU2r6fdWuNKKNBMBv QwIDAQAB

支付宝公钥:

复制公钥

MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAqVjSs4PrjOsJ/c/m7SDYoPKPIwvEwq3U6p2dzb0X/wVsYsZEHLrWEeVZfTHi9J0yrwnxgUPICGKv4EaJuc8msWuuH3TxEvurTPJxYUCOiBIQYeG5iRTh3fWYCKMILNm8UXv0v+KRVK9bZ17cWB0vBM7iwrytGbfKg0vhljsybbOd4VM/m4bWZOFdkXAmJfbO+pfMsa9TzF5zYmSpoHuBAkGu9m7EtzzBIXifhYkhvyEAX/XWG3rgaEe+I8W+NIPZ1uudZ2AIWLvnl5Jjrcz1yMJqOhsobBrPiWbnwnRsMz1Sn9W9rl/cGIcboPIEMgs988Vcd/o21gHDNwNKKQIDAQAB

查看接入文档

注意，支付宝公钥需要手动复制粘贴到文件中，步骤如下

1，用户目录下

```
vim alipay_public_key.pem
```

2，进入vim后手动添加如下两行

```
-----BEGIN PUBLIC KEY-----
```

```
-----END PUBLIC KEY-----
```

3，光标在BEGIN处 点击 键盘o 进入插入模式，此时光标停留在 BEGIN和END的两行之间；粘贴支付宝公钥 最终格式如下

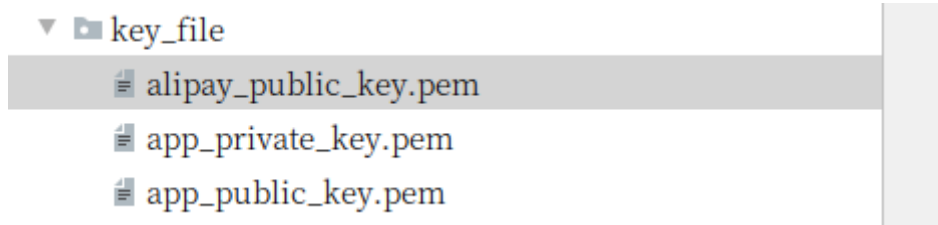
```
-----BEGIN PUBLIC KEY-----
```

```
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAg
VjSs4Prj0sJ/c/m7SDYoPKPIwvEwq3U6p2dzpB0X/wVsY
sZEHLrWEeVZfTHi9J0yrvwnxgUP1CGKv4EaJuc8msWuuH
3TxEvurTPJxYUCOiBIQYeG5iRTh3fwyCkM1LNm8UXv0v+
KRVK9bZ17cWB0vBM7iwrytGbfg0vhIjsybbOd4VM/m4b
WZOFdkXAmJfbO+pfMsa9TzF5zYmSpoHuBAkGu9m7EtzzB
1Xi fthYkhvyEAX/XWG3rgaEe+18W+N1PZ1uudZ2AIWLvn
L15Jjrcz1yMJqOhsobBrPiWbnwnRSMz1Sn9w9r1/cGITc
boP1EMgs988Vcd/o21gHDNwNKQIDAQAB
-----END PUBLIC KEY-----
```

4, `esc`退出插入模式, 执行 `:wq` 退出保存

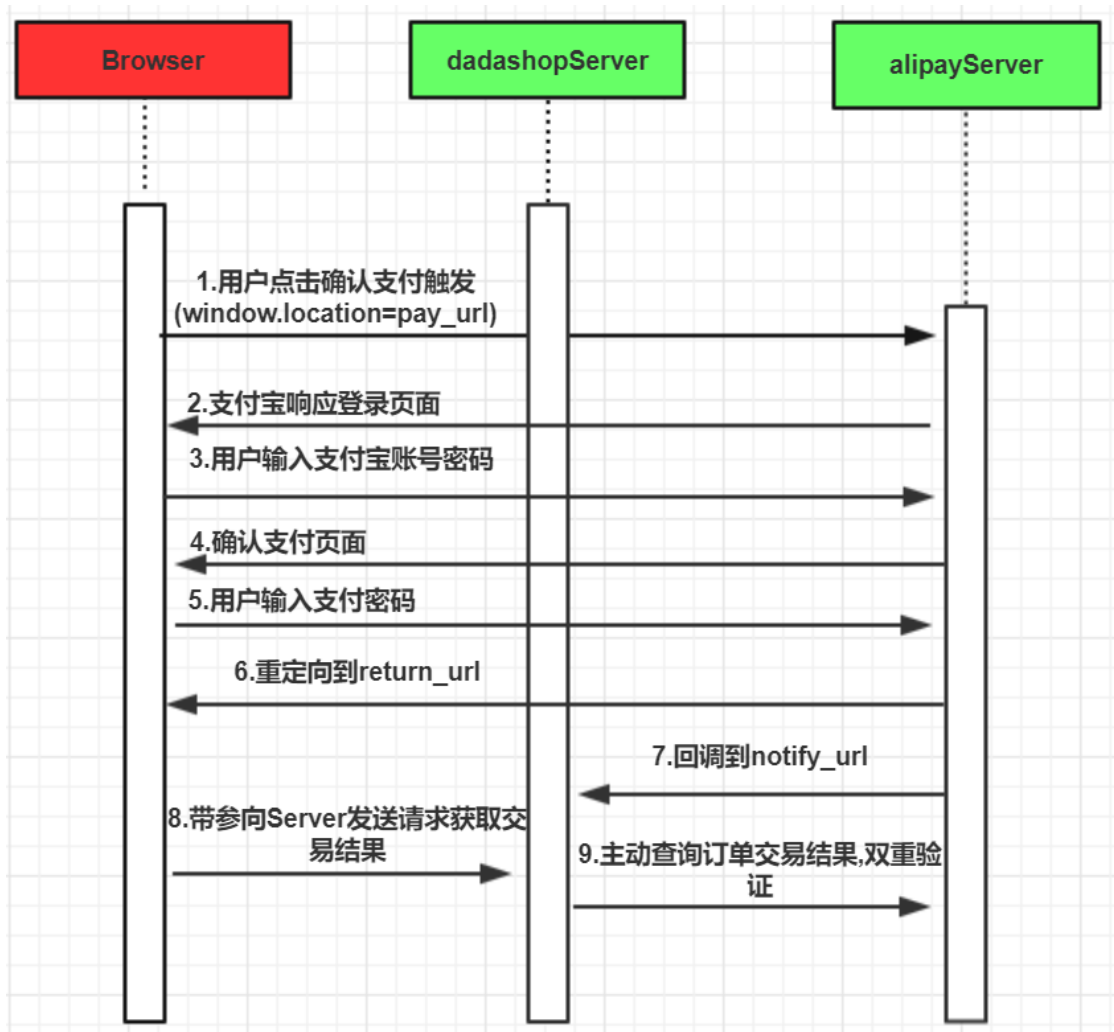
6.后端代码实现(Django):

- 6.1 公钥文件:需要将私钥和支付宝公钥 放到项目目录中.
 - 1.支付宝公钥
 - 2.应用私钥
 - 3.应用公钥



- 6.2 AliPay工具包:支付宝官方sdk没有针对Python的封装, 使用起来比较麻烦我们这里使用github码友封装的工具包安装详见(<https://github.com/fzlee/alipay/blob/master/README.zh-hans.md>)

支付时序图



alipay相关

○ 安装

安装python-alipay-sdk

```
sudo pip3 install python-alipay-sdk --upgrade
```

#安装成功后执行如下命令 校验安装结果

```
tarena@tedu:~$ pip3 freeze|grep -i ali
python-alipay-sdk==1.10.1 #输出此结果 则表示安装成功
```

○ 接口说明:

1.Alipay()

参数

```
# 应用id
appid="",
# 默认回调url，不写一定要用None填充
app_notify_url=None,
# 应用私钥
app_private_key_string="",
# 支付宝的公钥
alipay_public_key_string="",
# 签名方式 RSA或RSA2
sign_type="RSA",
# 运行环境
debug=False
```

返回值

一个支付对象

作用:

返回一个**支付对象**用作调用alipayAPI.

2.支付对象.api_alipay_trade_page_pay ()

参数

```
# 订单编号
out_trade_no="",
# 订单总金额
total_amount=0.01,
# 订单标题
subject=subject,
# 重定向URL
return_url="https://example.com",
# 回调URL, 可选! 不填则使用默认的notify_url
notify_url="https://example.com/notify"
```

返回值:

订单信息字符串.

作用:

与支付宝URL进行拼接,生成新的URL做跳转支付宝业务用.

3.支付对象.verify()

参数:

```
# 支付宝回调或重定向带回的参数, 具体参数如下框
result_dict
# 支付宝回调或重定向带回的签名
sgin
```

回调和重定向 参数 如下

charset=utf-8

out_trade_no=2019111821315602

method=alipay.trade.page.pay.return

total_amount=159.00

sign=BgLZAmnEnew1x%2BIJPx%2FWsvAj7xBupHQ
ksGXJlMfwhxKgJ2Fz1r4%2BL3cUXCOHwFKI7jOyB
nw%2F2aZTc84LuAXJZ2nD84HXuve04I%2FZlqUn7
Be5hMUVGPW8Itovoswy5srNwgpsNOzMqfLTO%2FP
GrCTFwwfJLqjHhCQVxGfg7CeUkEVy2wnjTTP1FxO
Ireckl5Ynqdy668SBADR1tQRTcc1Qj22fdH35hyt
wcbmQdufzNK%2FvGWldTcwzVRrFpMPZVjHs3399M
svrdrG2ASckXx6Y5QA5zouKGIVhnJzaM4ga3GgAv
zC2UwQEQ%2B2PzU8FMVqS1yNVNAQOBHUHqrJVzwa
cnQ%3D%3D

trade_no=2019120822001492241000052746

auth_app_id=2016100200644279

version=1.0

app_id=2016100200644279

sign_type=RSA2

seller_id=2088102178196953

timestamp=2019-12-08+16%3A34%3A36

返回值:

布尔类型 True 或 False

作用:

支付宝官方提供的验签.

4.支付对

象.api_alipay_trade_query(out_trade_no=order_id
)

参数:

```
# 订单编号  
out_trade_no
```

返回值:

订单支付的状态

作用:

主动查询支付结果

实现代码

```
from django.http import HttpResponse  
from django.shortcuts import render,  
redirect  
from django.views import View  
from alipay import AliPay  
from django import http  
import json  
import copy  
from django.conf import settings  
  
# Create your views here.  
# 本demo未配置数据库故订单id等使用常量表示  
TOTAL_AMOUNT = 159    # 订单金额[此金额需要后端  
通过数据查询]
```

```

ORDER_STATUS = 0          # 订单状态 0 初始状态 1
                            未付款 2 已付款
APPID = '2016101500689787'

# 处理跳转支付宝业务
class OrderProcessingnView(View):
    # 获取跳转支付页面
    def get(self, request):
        return
render(request, 'ajax_alipay.html')
    # 确认支付获取支付宝支付URL
    def post(self, request):
        json_obj = json.loads(request.body)
        #前端将订单号传至后端
        order_id = json_obj.get('order_id')
        alipay = AliPay(
            appid=APPID,
            app_notify_url=None, # 默认回调
url- 阿里与商户后台交互
            # 使用的文件读取方式,载入支付密钥

            app_private_key_path=settings.ALIPAY_KEY_D
IRS + 'app_private_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,
            # 使用文件读取的方式,载入支付宝公钥
            alipay_public_key_path=
settings.ALIPAY_KEY_DIRS +
            '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.alipaydev.com/gateway.do? +
order_string
    # 测试方式此为支付宝沙箱环境
    order_string =
alipay.api_alipay_trade_page_pay(
    out_trade_no=order_id,
    total_amount=TOTAL_AMOUNT,
    subject=order_id,
    # 回转url， 支付宝与买家业务处理完毕
(支付成功)将玩家重定向到此路由，带着交易的参数返回

    return_url="http://127.0.0.1:8000/payment/
result/",

    notify_url="http://127.0.0.1:8000/payment/
result/" # 可选，不填则使用默认notify url
)
    pay_url =
"http://openapi.alipaydev.com/gateway.do?"
+ order_string
    return
http.JsonResponse({"status":1,"pay_url":pay
_url})

# 处理支付宝回调及重定向业务
class PaymentResultView(View):
    # 获取参数字典和验签结果

```

```

def get_sdict_alipay_verify(self,
request_data):
    """
    :param request:
    :param method: 请求方式
    :return:
    success_dict, alipay_verify, alipay
    """

    success_dict =
copy.deepcopy(request_data)
    # 1. 剔除掉sign做验签准备
    sign = success_dict.pop("sign",
None)

    # 2. 生成alipay对象
    alipay = AliPay(
        appid=APPID,
        app_notify_url=None,

        app_private_key_path=settings.ALIPAY_KEY_D
IRS + 'app_private_key.pem',
        alipay_public_key_path=
settings.ALIPAY_KEY_DIRS +
'alipay_public_key.pem',
        debug=True
    )

    # 3. 使用支付宝接口进行验签
    alipay_verify =
alipay.verify(success_dict, sign)
    return success_dict, alipay_verify,
alipay

```



```
# 重定向接口
def get(self, request):
    # 1.获取参数字典,验签结果,alipay对象
    request_data = {k:request.GET[k]
for k in request.GET.keys()}
    print('request_data GET')
    print(request_data)
    success_dict, ali_verify, alipay =
self.get_sdict_ali_verify(request_data)
    # 2.根据验证结果进行业务处理
    if ali_verify is True:
        order_id =
success_dict.get('out_trade_no', None)
        if ORDER_STATUS == 2:
            return HttpResponse("订单支
付成功")

        # 主动查询
        else:
            result =
alipay.api_alipay_trade_query(out_trade_no=
order_id) # 主动查询接口
            if
result.get("trade_status", "") ==
"TRADE_SUCCESS":
                print('更改订单状态')
                #ORDER_STATUS = 2
                return HttpResponse("主
动查询结果订单支付完成了")
            else:
                return HttpResponse("支
付未完成")
```

```

        else:
            return HttpResponse("非法访问")

# 回调接口
def post(self, request):
    """
    处理支付宝的付款回调业务
    :param request:
    :return:
    """

    # 1. 获取参数字典, 验签结果, alipay对象
    request_data = {k:request.POST[k]
for k in request.GET.keys()}
    print('request_data POST')
    print(request_data)
    success_dict, ali_verify, alipay =
self.get_sdict_ali_verify(request_data)
    # 2. 根据验证结果进行业务处理
    if ali_verify is True:
        trade_status =
success_dict.get('trade_status', None)
        if trade_status ==
"TRADE_SUCCESS":
            print('更改订单状态')
            #ORDER_STATUS = 2
            return
HttpResponse("seccess")
        else:
            return HttpResponse("非法访问")

```

- 6.3 关于return_url和notify_url的问题
 - return_url 【GET】
 - 如果不给return_url支付宝处理完业务会留在自己的网页不做跳转
 - 重定向时会带上订单编号等参数
 - notify_url 【POST】
 - 支付结果异步通知
 - 对于 PC 网站支付的交易，在用户支付完成之后，支付宝会根据 API 中商户传入的 notify_url，通过 POST 请求的形式将支付结果作为参数通知到商户系统。
 - 详见(<https://docs.open.alipay.com/270/105902/>)

Demo 演示注意事项

启动demo后，访问 <http://127.0.0.1:8000/payment/jump/> 后可看到前端页面，点击页面按钮触发 支付