

微信支付 & 支付宝 — 完整对接指南

从 0 到 1，涵盖架构、流程、部署、测试、踩坑

一、软件架构

1.1 文件结构

```
webui/
├─ payment.py          # 支付核心模块
│   ├── WeChat Pay    # 微信 Native 扫码支付
│   │   ├── _get_client()      微信 SDK 客户端
│   │   ├── create_order()     套餐下单
│   │   ├── create_custom_order() 自定义金额下单
│   │   ├── verify_and_fulfill() 回调解密+验签+积分
│   │   └─ _save_order() / _get_order() 订单持久化
│   └─ Alipay          # 支付宝电脑网站支付
│       ├── _get_alipay()      支付宝 SDK 客户端
│       ├── create_alipay_order() 下单
│       └─ verify_alipay_notify() 回调验签+积分
├─ app.py              # Flask 路由层
│   ├── GET /recharge      充值页面
│   ├── POST /api/payment/create 微信套餐下单
│   ├── POST /api/payment/custom 微信自定义下单
│   └─ POST /api/payment/alipay  支付宝下单
```

```

|   |— POST /api/payment/wx_notify      微信回调
|   |— POST /api/payment/alipay_notify  支付宝回调
|— template.html      # 前端充值页面
|   |— selectPlan()   套餐选择
|   |— createPayment() 微信支付
|   |— createAlipayOrder() 支付宝支付
|   |— createCustomPayment() 自定义金额支付
|— db.py              # 数据库层 (users / pay_orders / usage_logs)

```

1.2 数据库表

pay_orders — 支付订单表

```

trade_no TEXT PRIMARY KEY  # 订单号 (WFC/Alipay前缀)
uid INTEGER                # 用户ID
points INTEGER              # 积分
expires REAL                # 过期时间
fulfilled INTEGER DEFAULT 0 # 0=待支付 1=已到账

```

usage_logs — 操作日志表

```

user_id, action, points_change, detail, created_at

```

二、交互流程

2.1 充值完整流程

```
用户 → https://qianshi.cool/recharge
|
├─ 选套餐 → 点「微信」或「支付宝」
|   └─ selectPlan(idx, type)
|       └─ type=wechat → createPayment() → POST /api/payment/create
|       └─ type=alipay → createAlipayOrder() → POST /api/payment/alipay
|
├─ 自定义金额 → 点「微信」或「支付宝」
|   └─ createCustomPayment(type)
|       └─ type=wechat → POST /api/payment/custom
|       └─ type=alipay → POST /api/payment/alipay
|
▼
后端 payment.py
|
├─ 微信
|   └─ _get_client() → WeChatPay SDK
|   └─ client.pay() → 微信返回 code_url
|   └─ _save_order() → 订单写入 SQLite
|   └─ 前端用 api.qrserver.com 将 code_url 生成 QR 码图片
|   └─ 用户微信扫码支付
|   └─ 微信 POST 回调 → /api/payment/wx_notify
|   └─ verify_and_fulfill() → AESGCM 解密 resource
|   └─ _get_order() → 查订单
|   └─ update_points() → 加积分
|   └─ add_log() → 记录日志
|
├─ 支付宝
|   └─ _get_alipay() → AliPay SDK
|   └─ api_alipay_trade_page_pay() → pay_url
|   └─ _save_order() → 订单写入 SQLite
|   └─ 前端 window.open(pay_url) → 支付宝收银台
|   └─ 用户完成支付
|   └─ 支付宝 POST 回调 → /api/payment/alipay_notify
|   └─ verify_alipay_notify() → RSA2 验签
|   └─ _get_order() → 查订单
```

```
└─ update_points() → 加积分  
└─ add_log() → 记录日志
```

2.2 微信 QR 码生成

```
// 后端返回 code_url: "weixin://wmpay/bizpayurl?pr=xxx"  
let r = await fetch('/api/payment/custom', {  
  method: 'POST', body: JSON.stringify({amount: 1})  
});  
let j = await r.json();  
// 通过 Google QR API 生成二维码图片  
let qrUrl = 'https://api.qrserver.com/v1/create-qr-code/?size=200x200&data='  
  + encodeURIComponent(j.qr_url);  
document.getElementById('qrCode').innerHTML  
  = '';
```

2.3 支付宝支付跳转

```
let r = await fetch('/api/payment/alipay', {  
  method: 'POST', body: JSON.stringify({amount: 1})  
});  
let j = await r.json();  
// 新窗口打开支付宝收银台  
window.open(j.pay_url);
```

三、安装与部署

3.1 安装依赖

```
# 微信支付 SDK
pip3 install --break-system-packages wechatpayv3

# 支付宝 SDK
pip3 install --break-system-packages python-alipay-sdk

# 微信回调解密 (必须)
pip3 install --break-system-packages cryptography
```

3.2 微信支付 — 平台配置

步骤	操作	说明
---	---	---
1	pay.weixin.qq.com 申请商户号	普通商户，选 PC 网站场景
2	产品中心 → Native 支付 → 开通	
3	API 安全 → 下载 API 证书	得 apiclient_key.pem
4	API 安全 → 设置 APIv3 密钥	32 位自编
5	API 安全 → 微信支付公钥	拿 pub_key_id + pub_key.pem
6	产品中心 → AppID 账号管理	绑定小程序/服务号 AppID

3.3 支付宝 — 平台配置

步骤	操作	说明
---	---	---
1	open.alipay.com 创建网页应用	拿 APPID
2	接口加签 → 下载支付宝助手生成 RSA2 密钥	上传公钥，保存私钥
3	b.alipay.com 签约电脑网站支付	
4	应用上线	状态变为「已上线」

3.4 代码配置

payment.py 修改以下变量：

```
# 微信
WECHAT_MCHID = "你的商户号"
WECHAT_APPID = "你的AppID"
WECHAT_API_V3_KEY = "你的32位密钥"
WECHAT_CERT_SERIAL = "证书序列号"
WECHAT_PUB_KEY_ID = "公钥ID"
CALLBACK_HOST = "https://你的域名"

# 支付宝
ALIPAY_APPID = "你的APPID"
```

```
ALIPAY_PRIVATE_KEY = ""应用私钥""
ALIPAY_PUBLIC_KEY = ""支付宝公钥""
```

3.5 证书文件

```
certs/
├── 1115016766/
│   ├── apiclient_key.pem      # 微信商户私钥
│   ├── apiclient_cert.pem    # 微信商户证书
│   └── pub_key.pem            # 微信支付公钥
zhifubao/
└── readme.md                  # 支付宝密钥
```

3.6 Nginx 配置

确保 `/api/` 路径被代理到 Flask:

```
location /api/ {
    proxy_pass http://127.0.0.1:8888/api/;
    proxy_set_header Host $http_host;
    proxy_set_header X-Real-IP $remote_addr;
}
```

四、测试

4.1 后端测试

```
# 登录获取 session
curl -X POST http://127.0.0.1:8899/api/auth/login \
  -H 'Content-Type: application/json' \
  -d '{"email":"admin@qianshi.cool","password":"admin123"}' -c /tmp/t.txt

# 微信 0.1 元测试
curl -X POST http://127.0.0.1:8899/api/payment/custom \
  -b /tmp/t.txt -H 'Content-Type: application/json' \
  -d '{"amount":0.1}'
# → {"ok":true, "qr_url":"weixin://wxpay/bizpayurl?pr=xxx"}

# 支付宝 0.1 元测试
curl -X POST http://127.0.0.1:8899/api/payment/alipay \
  -b /tmp/t.txt -H 'Content-Type: application/json' \
  -d '{"amount":0.1}'
# → {"ok":true, "pay_url":"https://openapi.alipay.com/..."}
```

4.2 前端测试

打开 `https://你的域名/recharge` :

1. 输入 0.1 元 → 点微信 → 扫码支付 → 检查积分是否增加
2. 输入 0.1 元 → 点支付宝 → 完成支付 → 检查积分是否增加

4.3 回调验证

```
# 查订单
sqlite3 data/app.db "SELECT * FROM pay_orders ORDER BY rowid DESC LIMIT 5"

# 查日志
sqlite3 data/app.db "SELECT * FROM usage_logs WHERE action='充值' ORDER BY id DESC LIMIT 5"

# 查积分
sqlite3 data/app.db "SELECT email,points FROM users"
```

4.4 成功标志

- `pay\_orders.fulfilled = 1`
- `usage\_logs` 有「充值」记录
- 用户积分增加

五、踩坑记录

微信支付

问题	原因	解决
---	---	---
回调 trade_no 为空	v3 回调 resource 字段 AEAD_AES_256_GCM 加密	手写 `AESGCM` 解密
SDK callback() 报错	内部验签与回调格式不兼容	弃用 SDK，手动解密

金额小数被拒	<code>`int(0.1)` → 0</code>	<code>`float()`</code>
本地充值无记录	回调 URL 指向云端，数据库分离	必须走 <code>`https://域名/recharge`</code>
gunicorn 日志无输出	<code>`logging.getLogger('app')`</code> 不到 gunicorn	<code>`sys.stderr.write()`</code>

支付宝

问题	原因	解决
---	---	---
insufficient-isv-permissions	签约后权限未刷新	新建应用重新签约
not-online-app	应用未发布	点击上线
验签失败	<code>`request.form`</code> 修改参数	<code>`request.get_data()`</code> 拿原始 body
金额小数报错	<code>`int()`</code> 截断	<code>`float()`</code>

六、充值规则

- 最小 0.1 元，最大 10000 元
- 支持 2 位小数
- 1 元 = 10 积分
- 套餐：10 元/100 分 ~ 1000 元/12500 分

