

在线支付

向军大叔每晚八点在 [抖音](#) 和 [bilibili](#) 直播



[微信支付](#)是腾讯公司的支付业务品牌,微信支付商户平台支持线下场所、公众号、小程序、PC 网站、APP、企业微信等经营场景快速接入微信支付。

支付配置

下面通过 PC 扫码支付来演示微信支付的使用，其他支付方式与此大同小异。

首先需要在[微信支付平台](#)申请帐号，微信提供多款[支付产品](#)用于满足不同应用场景的支付。你可以通过[微信支付平台](#)了解更多细节。

支付产品



JSAPI支付
商户调用JSAPI接口，调起微信支付模块收款



APP支付 ! 未开通
用户在APP中发起支付



Native支付
商户系统按微信支付协议生成支付二维码，用户扫码支付



付款码支付
用户打开微信钱包付款卡页面，商户扫码完成支付



H5支付
在微信客户端外的移动端网页使用微信支付



刷脸支付 ! 未开通
用户在支持微信刷脸支付的机器上，可以刷脸完成支付

产品	说明
JSAPI 支付	使用微信 APP 内部浏览器发起支付，适用于微信公众号等场景
APP 支付	在 APP 中发起支付
Native 支付	PC 网页生成二维码，用户使用微信扫码支付
H5 支付	手机端浏览器的 H5 页面中发起支付，会调取用户微信 APP 完成支付

API 证书

访问微信支付[帐户中心](#) 并下载证书工具来生成证书。

每年生成证书有次数限制，所以生成好的证书请妥善保管



执行软件首先设置证书保存目录



设置商户信息（初次执行会自动填写）

微信支付商户平台证书工具 V1.4

— ×

● 填写商户信息

● 复制请求串

● 粘贴证书串

● 生成证书

填写商户信息并点击下一步获取证书

粘贴

商户号

1:

请输入开户邮件中的商户号

商户名称

北京后

商户名须与营业执照名称一致

下一步

上一步

houdunren.com@向军老师

然后复制软件第二步生成的请求串

微信支付商户平台证书工具 V1.4

— ×

✓ 填写商户信息

● 复制请求串

● 粘贴证书串

● 生成证书

请点击右侧“复制”按钮，将证书请求串复制至【商户平台】

-----BEGIN CERTIFICATE REQUEST-----
MIIC5zCCAC8CAQAwgaExCzAJBgNVBAYTAkNOMRIwEAYDVQQIDAIHdWFluZ0Rvbmcx
E
r
(
-
C

复制

请完成粘贴至【商户平台】后，再点击下一步按钮

下一步

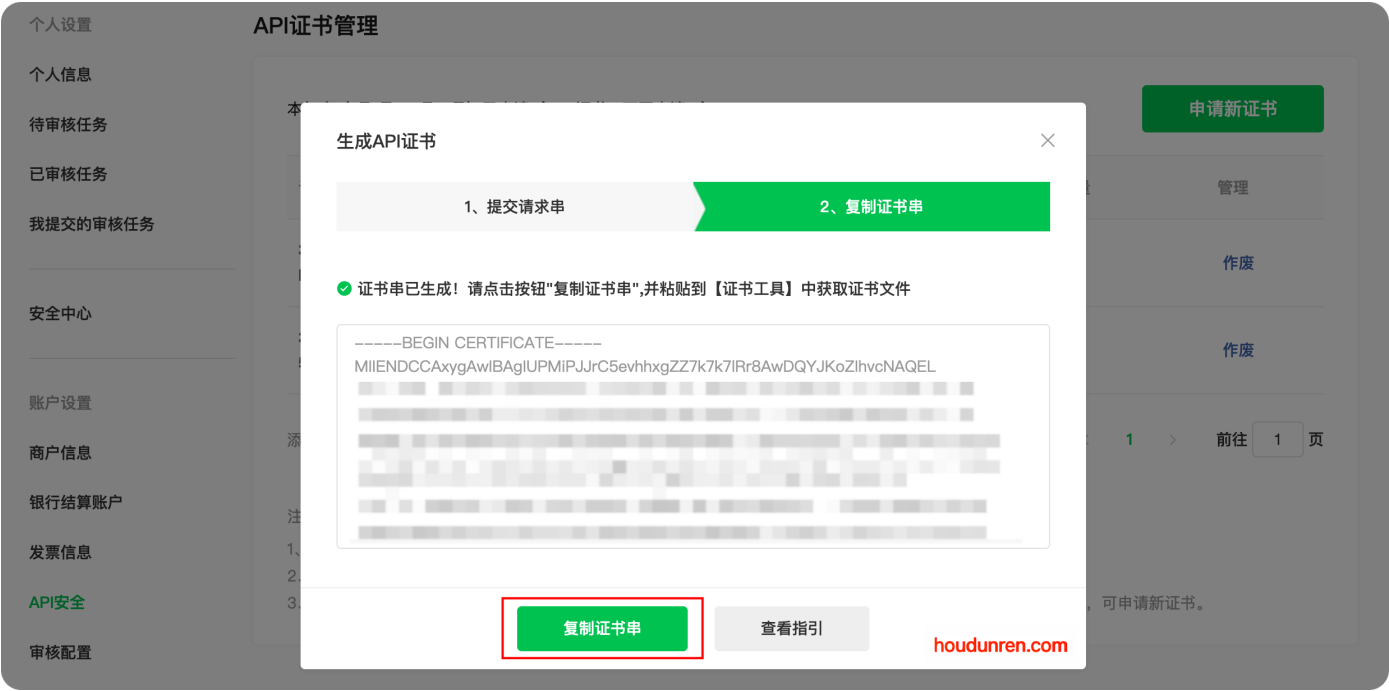
上一步

houdunren.com

接着回到微信支付网站，将复制的请求串粘贴



接着点击下一步，然后复制证书串



然后回到软件中，将上一步复制的证书串粘贴（软件会自动粘贴）

✓ 填写商户信息 ——— ✓ 复制请求串 ——— ● 粘贴证书串 ——— ● 生成证书

请将证书串粘贴至文本框后点击“下一步”按钮

-----BEGIN CERTIFICATE-----
MIENDCCAxygAwIBAgIUPMiPJrC5evhhxgZZ7k7k7lRr8AwDQYJKoZIhvcN
AQEL

粘贴

完成粘贴后，点击“下一步”按钮，即可获得证书文件和私钥文件

下一步

上一步

houdunren.com

然后完成证书的申請，点击 [查看证书文件夹](#) 链接

✓ 填写商户信息 ——— ✓ 复制请求串 ——— ✓ 粘贴证书串 ——— ● 生成证书



申请证书成功

电脑本地已生成证书文件，[点击查看证书文件夹](#)

houdunren.com

将最终生成的证书妥善保存

个人收藏

桌面

影片

图片

文稿

共享文件夹

apiclient_cert.p12

apiclient_cert.pem

apiclient_key.pem

houdunren.com

API 密钥

回到微信支付网站设置 32 位 API 密钥，可以使用[随机密码生成器](#)来生成，并设置到微信支付的[API 安全](#)中，并妥善保存以供后面程序中使用。

API安全 1

审核配置

微信群管理

员工账号管理

刷脸智能设备

微信商户终端

微信支付物料管理

支付申请

消费者投诉

违约记录

微信支付公钥 未启用 新

微信支付回调、应答商户时会使用微信支付私钥签名，商户需要使用微信支付公钥验签，用于验证微信支付的身份

申请公钥 查看指引

平台证书 已申请 2029年11月16日过期

商户调用业务API后，微信支付回调会使用平台证书的私钥生成签名，商户需要使用平台证书的公钥验签。平台证书5年过期一次，为保证业务正常使用，请及时更换

管理证书 查看指引

解密回调 2

商户解密APIv3的回调通知使用

APIv3密钥 已申请 2024年11月17日修改


调用APIv3的下载平台证书接口、处理回调通知中报文时，要通过该密钥来解密信息（APIv3密钥与APIv2密钥互不影响）

修改 查看指引

houdunren.com

保存好 APIv3密钥，后面在网站程序中要使用

验证微信支付身份

微信支付

商户平台

首页

交易中心

账户中心1

营销中心

产品中心

数据中心

个人设置

个人信息

待审核任务

已审核任务

我提交的审核任务

安全中心

账户设置

商户信息

银行结算账户

发票信息

API安全2

审核配置

微信群管理

员工账号管理

刷脸智能设备

微信商户终端

API安全

验证商户身份

商户调用API时签名使用

商户API证书已申请2029年11月16日过期

商户API证书用于证实商户身份，需妥善保管防止泄露：
APIv2中，调用微信支付安全级别较高的接口；APIv3中，调用微信支付所有接口需使用

管理证书查看指引

商户APIv2密钥未设置

调用APIv2时，需用APIv2密钥签名，从而界定你的身份，需妥善保管防止泄露（APIv2密钥与APIv3密钥互不影响）

设置查看指引

验证微信支付身份

商户接收APIv3的请求应答、回调时验签使用，以下两种方式只能使用一种

微信支付公钥切换中3新

微信支付回调、应答商户时会使用微信支付私钥签名，商户需要使用微信支付公钥验签，用于验证微信支付的身份

查看进度查看指引

平台证书已申请2029年11月16日过期

商户调用业务API后，微信支付回调会使用平台证书的私钥生成签名，商户需要使用平台证书的公钥验签。平台证书5年过期一次，为保证业务正常使用，请及时更换

管理证书查看指引

houdunren.com

然后下载公钥文件并获取公钥 ID，这会在下面代码的微信支付配置项的 `wechat_public_cert_path` 中使用

我提交的审核任务

安全中心

账户设置

商户信息

银行结算账户

发票信息

API安全

第一步 下载微信支付公钥1

下载的文件中包含公钥 ID 和公钥

重新下载2

公钥ID: PUB_KEY复制3

第二步 按照指引完成开发

1) 回调场景：微信支付会将签名所用平台证书序列号或微信支付公钥ID放入回调请求的HTTP头部 Wechatpay-Serial中。
2) 请求-应答场景：商户应将所持有的平台证书序列号或微信支付公钥ID放入请求的HTTP头部Wechatpay-Serial中。
商户应做好兼容验签，详情见：开发指引

查看指引

houdunren.com

商户号

1. 登录微信支付[产品中心](#) 选择支付产品



2. 点击 **Native支付**，然后在「开发配置」中查看商户号



Native 支付

Native 用于在 PC 桌面端生成二维码完成支付。

绑定服务号

使用 [Native 支付](#) 需要首先绑定具有支付功能的微信服务号。请在 [AppID 帐号管理](#) 中绑定微信服务号。

产品大全

我的产品

开发配置

AppID账号管理

AppID账号管理

我关联的AppID账号

① 完成关联的AppID账号，将可用于支付，营销，用户触达，支付凭证小程序等业务。 [查看指引](#)

请输入昵称/AppID

搜索

+ 关联AppID

昵称	AppID	账号类型	关联状态	操作
后盾云	wxc41245	服务号	已关联	houdunren.com@向军大叔

设置回调地址

在 产品中心 > [开发配置](#) > 支付配置 > Native支付中设置Native支付回调链接

Native支付

Native支付回调链接

Native支付回调链接

操作

<https://www.houdunren.com>[修改](#)

h5 支付

使用 h5 支付时即在移动端浏览器支付，也需要绑定具有支付功能的微信服务号。

配置支付域名

然后访问 产品中心 -> [开发配置](#) -> 支付配置->JSAPI支付，添加支付授权目录

支付配置

JSAPI支付

支付授权目录 [添加](#)

JSAPI支付授权目录（最多可添加5个）

操作

<https://www.>[删除](#)<https://www>[删除](#)[https://www.](https://www)[删除](#)[houdunren.com](https://www.houdunren.com)

程序编码

下面来使用 [yansongda/pay](#) 快速为网站集成微信 **Native**支付。

相比自己完成支持业务来讲，使用扩展包快速便捷。

- 使用的金额单位为 分

参数说明

下面对开发中用到的参数进行说明

1. mch_id 微信支付商户号 [帐户中心>商户信息](#)
2. key 商户支付 API 密钥 [帐户中心>API 安全>解密回调](#) 要使用 **APIv3** 版本的（上面我们已经生成了）

支付申请

消费者投诉

违约记录

解密回调

商户解密APIv3的回调通知使用

APIv3密钥

已申请

调用APIv3的下载平台证书接口、处理回调通知中报文时，要通过该密钥来解密信息（APIv3密钥与APIv2密钥互不影响）

修改

查看指引

houdunren.com

3. app_id 公众号应用 ID [产品中心>AppID 账号管理](#)

微信支付

商户平台

首页

交易中心

账户中心

营销中心

1 产品中心

数据中心

产品大全

我的产品

开发配置

AppID账号管理 2

AppID账号管理

我关联的AppID账号

完成关联的AppID账号，将可用于支付，营销，用户触达，支付凭证小程序等业务。

查看指引

请输入昵称/AppID

搜索

+ 关联AppID

昵称	AppID	账号类型	关联状态	操作
3	4 wxf5	服务号	已关联	查看

安装扩展

使用 composer 安装扩展包

```
composer require yansongda/pay -vvv
```

安装 [endroid/qrcode](#) 用于生成二维码

```
composer require endroid/qrcode
```

配置项

创建配置文件 **config/pay.php** 内容如下

```
<?php

use Yansongda\Pay\Pay;

return [
    'alipay' => [
        'default' => [
            // 必填-支付宝分配的 app_id
            'app_id' => env('ALIPAY_APP_ID'),
            // 必填-应用私钥 字符串或路径
            'app_secret_cert' => env('ALIPAY_APP_SECRET_CRET'),
            // 必填-应用公钥证书 文件类似:appCertPublicKey_2017xxxx.crt
            'app_public_cert_path' => base_path(env('ALIPAY_APP_PUBLIC_CERT_PATH')),
            // 必填-支付宝公钥证书 文件类似:ertPublicKey_RSA2.crt
            'alipay_public_cert_path' => base_path(env('ALIPAY_PUBLIC_CERT_PATH')),
            // 必填-支付宝根证书 文件类似:crt/alipayRootCert.crt
            'alipay_root_cert_path' => base_path(env('ALIPAY_ROOT_CERT_PATH')),
            'return_url' => '',
            'notify_url' => '',
            // 选填-第三方应用授权token
            'app_auth_token' => '',
            // 选填-服务商模式下的服务商 id, 当 mode 为 Pay::MODE_SERVICE 时使用该参数
            'service_provider_id' => '',
            // 选填-默认为正常模式。可选为: MODE_NORMAL, MODE_SANDBOX, MODE_SERVICE
            'mode' => Pay::MODE_NORMAL,
        ]
    ],
    'wechat' => [
        'default' => [
            // 必填-商户号, 服务商模式下为服务商商户号
            'mch_id' => env("WEPAY_MCH_ID"),
            // 必填-商户秘钥 key 访问微信支付网站 [帐户中心>API 安全>设置 API 密钥], 要使用
            // **APIv3** 版本的
            'mch_secret_key' => env('WEPAY_MCH_SECRET_KEY'),
            // 必填-商户私钥 字符串或路径 微信支付证书工具生成的 apiclient_key.pem 文件
            'mch_secret_cert' => base_path('pay/wepay/apiclient_key.pem'),
            // 必填-商户公钥证书路径 微信支付证书工具生成的 apiclient_cert.pem 文件
            'mch_public_cert_path' => base_path('pay/wepay/apiclient_cert.pem'),
            // 必填
            'notify_url' => '',
            // 选填-公众号 的 app_id
            'mp_app_id' => 'wxc47243ed572e273d',
            // 选填-小程序 的 app_id
            'mini_app_id' => '',
            // 选填-app 的 app_id
```

```

        'app_id' => '',
        // 选填-合单 app_id
        'combine_app_id' => '',
        // 选填-合单商户号
        'combine_mch_id' => '',
        // 选填-服务商模式下, 子公众号 的 app_id
        'sub_mp_app_id' => '',
        // 选填-服务商模式下, 子 app 的 app_id
        'sub_app_id' => '',
        // 选填-服务商模式下, 子小程序 的 app_id
        'sub_mini_app_id' => '',
        // 选填-服务商模式下, 子商户id
        'sub_mch_id' => '',
        // 选填-微信公钥证书路径, optional, 强烈建议 php-fpm 模式下配置此参数
        'wechat_public_cert_path' => [
            //'45F59D4DABF31918AFCEC556D5D2C6E376675D57' => __DIR__ .
            '/Cert/wechatPublicKey.crt',
            //上面生成的微信支付公钥
            'PUB_KEY_ID_011699xxxxxxxxx6' => base_path('pay/wepay/publickey.pem'),
        ],
        // 选填-默认为正常模式。可选为: MODE_NORMAL, MODE_SERVICE
        'mode' => Pay::MODE_NORMAL,
    ]
},
'unipay' => [
    'default' => [
        // 必填-商户号
        'mch_id' => '777290',
        // 必填-商户公私钥
        'mch_cert_path' => __DIR__ . '/Cert/unipayAppCert.pfx',
        // 必填-商户公私钥密码
        'mch_cert_password' => '000000',
        // 必填-银联公钥证书路径
        'unipay_public_cert_path' => __DIR__ . '/Cert/unipayCertPublicKey.cer',
        // 必填
        'return_url' => 'https://yansongda.cn/unipay/return',
        // 必填
        'notify_url' => 'https://yansongda.cn/unipay/notify',
    ],
],
'logger' => [
    'enable' => false,
    'file' => base_path('logs/pay.log'),
    'level' => 'info', // 建议生产环境等级调整为 info, 开发环境为 debug
    'type' => 'single', // optional, 可选 daily.
    'max_file' => 30, // optional, 当 type 为 daily 时有效, 默认 30 天
],
'http' => [ // optional
    'timeout' => 5.0,
    'connect_timeout' => 5.0,
    // 更多配置项请参考 [Guzzle](https://guzzle-cn.readthedocs.io/zh_CN/latest/request-
options.html)

```

```
],  
];
```

业务处理

下面是 Laravel 框架为例编写后台业务代码。创建控制器 App\Http\Controllers\PayController.php 内容如下

- PC 扫码支付需要安装 [qr-code](#) 扩展来生成二维码

```
<?php  
  
namespace App\Http\Controllers;  
  
use App\Models\Order;  
use Illuminate\Http\Request;  
use Log;  
use Yansongda\Pay\Pay;  
use App\Services\OrderService;  
use Endroid\QrCode\Builder\Builder;  
  
class WepayController extends Controller  
{  
    public function __construct()  
    {  
        $config = config('pay');  
        $config['wechat']['default']['notify_url'] = url('/wepay/notify');  
        Pay::config($config);  
    }  
  
    //发起支付  
    public function pay()  
    {  
        $order = [  
            'out_trade_no' => time(),  
            'description' => '后盾人视频',  
            'amount' => [  
                'total' => 1,  
            ],  
        ];  
        $result = Pay::wechat()->scan($wechatOrder);  
        //生成二维码图片  
        $builder = new Builder(  
            writer: new PngWriter(),  
            writerOptions: [],  
            validateResult: false,  
            data: $result['code_url'],  
            encoding: new Encoding('UTF-8'),  
            errorCorrectionLevel: ErrorCorrectionLevel::High,  
            size: 300,  
            margin: 0,
```

```

        roundBlockSizeMode: RoundBlockSizeMode::Margin,
        // logoPath: url('/images/xj.jpeg'),
        logoResizeToWidth: 50,
        logoPunchoutBackground: true,
    );
    header("Content-Type:image/png");
    return $builder->build()->getString();
}

//异步通知
public function async()
{
    $pay = Pay::alipay();
    $result = $pay->callback();
    Log::info((array)$result);
    //修改订单状态...
    return $pay->success();
}
}

```

路由配置

下面来配置路由，要保证最终路由是以 https:// 开始的，否则支付成功的通知事件不会触发

```

Route::get('wepay/pay',[WepayController::class,'pay']);
Route::post('wepay/pay',[WepayController::class,'async']);

```

前台使用

前台发送 axios 请求，需要指定响应类型为 blob，下面是 react 代码，其他框架自行调整

```

const res = (await axios.post('/wepay/pay', {}, {
  responseType: 'blob'
})).data

<img src={URL.createObjectURL(res)}/>

```