

# 分享-APP支付宝支付开发

## 一. 申请支付宝商户功能

1.进入支付宝商户平台， 点击app支付或者即时到账



2.点击立即签约



3. 填写资料然后签约

## 二、集成支付宝sdk

打开<https://doc.open.alipay.com/docs/doc.htm?spm=a219a.7629140.0.0.NBPh2F&treeld=193&articleId=104511&docType=1>

# 其他产品功能

更新时间：2016/06/08    访问次数：87224

手机网站支付	DEMO
即时到账	DEMO
快捷登录	DEMO
批量付款到支付宝账户	DEMO
即时到账有密退款	DEMO
报关	DEMO

## FAQ

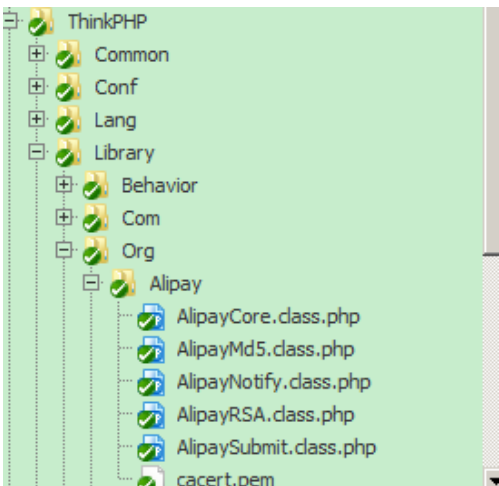
点击即时到账

即时到账交易接口(create\_direct\_pay\_by\_user) ( DEMO下载 )  
即时到账批量退款有密接口(refund\_fastpay\_by\_platform\_pwd) ( DEMO下载 )

解压下载的demo，找到php版本的，把lib文件夹下的文件集成到系统中



我已经集成到系统的ThinkPHP/Library/Org/Alipay下面了



### 三、配置支付宝conf文件

1. 到Application/Common/Conf/config.php,里面配置下alipay\_config和alipay, 里面需要填写partner,key, sign\_type, input\_charset, notify\_url等

其中partner和key请登录支付宝商户平台,

到<https://b.alipay.com/order/serviceIndex.htm>页面点击查询PID、Key来进行查询, 如下图:



### 四、系统内使用支付宝接口

1.配置参数, 产生请求字符串, 如果是APP端的请求会返回参数orderInfo, app端通过参数调用支付宝支付, 如果是网页版会直接跳转到支付宝支付页面。

//构造要请求的参数数组, 无需改动

```
$parameter = array(

    "service" => $service,

    "partner" => trim($alipay_config['partner']),

    "payment_type" => $payment_type,

    "notify_url" => $notify_url,

    "return_url" => $return_url,

    "seller_id" => $seller_id,

    "out_trade_no" => $out_trade_no,

    "subject" => $subject,

    "total_fee" => $total_fee,

    "body" => $body,

    "exter_invoke_ip" => $exter_invoke_ip,

    "_input_charset" => trim(strtolower($alipay_config['input_charset']))

);
```

```

//建立请求

$alipaySubmit = new AlipaySubmit($alipay_config);

$this->trackLog("alipaySubmit", $alipaySubmit);

if($alipayType == AlipayTypeEnum::TYPE_APP){

    //拼接订单字符串，如果是APP端请求，返回字符串由apicloud调用支付宝完成

    $orderInfo = $alipaySubmit->buildRequestParaToString($parameter);

    $this->trackLog("orderInfo", $orderInfo);

    $data['code'] = 1;

    $data['msg'] = "生成支付参数成功! ";

    $data['orderInfo'] = $orderInfo;

    $data['orderId'] = $walletOrderId;

    $this->ajaxReturn($data);

}else if($alipayType == AlipayTypeEnum::TYPE_PC){

    //如果是pc端请求，直接发起网页支付

    $html_text = $alipaySubmit->buildRequestForm($parameter,"post", "确认");

    echo $html_text;

}

```

## 2.App端通过前台返回的参数调用支付宝支付

```

function recharge(){

    console.log("recharge");

    var token = $api.getStorage('token');

    $.ajax({

        type: 'get',

```

```
url: C_DATA_SERVER_URL + 'Wallet/confirmRecharge',
```

```
data: {
```

```
    'token': token,
```

```
    'amount': '0.01'
```

```
},
```

```
dataType: 'json',
```

```
async: false,
```

```
timeout: 3000,
```

```
context: $(''.wallet-bg h1'),
```

```
success: function (res) {
```

```
    api.hideProgress();
```

```
    if (res.code === 1) {
```

```
        console.log("success");
```

```
        var aliPay = api.require('aliPay');
```

```
        aliPay.payOrder({
```

```
            orderInfo: res.orderInfo
```

```
        }, function(ret, err) {
```

```
            api.alert({
```

```
                title: '支付结果',
```

```
                msg: ret.code,
```

```
                buttons: ['确定']
```

```
                //code = 9000 && success=true才能判断成功
```

```
            });
```

```
        });
```

```

    } else {

        $(this).html("--.--");

        myToast(res.msg);

    }

},

error: function () {

    api.hideProgress();

    server_error();

}

});

}

```

(3) 后台收到支付宝的异步通知，获取支付结果并完成订单操作

```

public function notify(){

    $this->trackLog("execute", "notify()");

    //计算得出通知验证结果

    $alipay_config=C('alipay_config');

    $this->trackLog("trade_status", $_POST['trade_status']);

    $this->trackLog("post.", I('post.'));

    $alipayNotify = new AlipayNotify($alipay_config);

    $verify_result = $alipayNotify->verifyNotify();

    $this->trackLog("$verify_result", $verify_result);

    if(!$verify_result) {

```

```
//验证失败
```

```
$this->trackLog("notify verify", "fail");
```

```
echo "fail";
```

```
return;
```

```
}
```

```
//验证成功
```

```
if($_POST['trade_status'] == 'TRADE_FINISHED') {
```

```
    //这里是担保交易真正交易成功了，钱打入商家的账户
```

```
    //目前不是担保交易所以这里不处理
```

```
}else if ($_POST['trade_status'] == 'TRADE_SUCCESS') {
```

```
    //买家的钱已经进入支付宝平台或者商户账户（即时到账）
```

```
    //目前平台使用即时到账接口，所以在这里处理订单
```

```
    //获取支付宝的通知返回参数，可参考技术文档中服务器异步通知参数列表
```

```
    $outTradeNo    = $_POST['out_trade_no'];    //商户订单号
```

```
    $transactionId = $_POST['trade_no'];        //支付宝交易号
```

```
    $tradeStatus   = $_POST['trade_status'];    //交易状态
```

```
    $totalFee      = $_POST['total_fee'] * 100; //交易金额 支付宝返回的小数，平台存到的是为分的整数，所以要乘以100
```

```
    $notifyId      = $_POST['notify_id'];        //通知校验ID。
```

```
    $notifyTime    = $_POST['notify_time'];      //通知的发送时间。格式为yyyy-MM-dd
```

HH:mm:ss。

```
$buyerEmail = $_POST['buyer_email'];    //买家支付宝帐号;
```

```
$status = WalletEnum::STATUS_ORDER_SUCCEEDED;
```

```
$this->trackLog("outTradeNo", $outTradeNo);
```

```
$this->trackLog("transactionId", $transactionId);
```

```
$this->trackLog("totalFee", $totalFee);
```

```
$this->trackLog("status", $status);
```

```
//开启事务处理：处理完成之后更新订单号状态，给用户加钱
```

```
$db = new Model();
```

```
$db->startTrans();
```

```
$walletOrderService = new WalletOrderService($db);
```

```
$walletService = new WalletService($db);
```

```
//检查订单状态，是否为处理中
```

```
$curStatus = $walletOrderService->getStatusByMD5Id($outTradeNo);
```

```
if($curStatus == WalletEnum::STATUS_ORDER_PROCESSING) {
```

```
    //更新订单状态
```

```
    $statusResult = $walletOrderService-  
>updateStatusAndOrderIdByMD5Id($outTradeNo, $transactionId, $status);
```

```
    $this->trackLog("statusResult", $statusResult);
```

```
    //给用户加钱
```

```
$walletId = $walletOrderService->getToWalletIdByMD5Id($outTradeNo);
```

```
$addResult = $walletService->addMoney($walletId, $totalFee);
```



```
$this->trackLog("walletId", $walletId);

$this->trackLog("statusResult", $statusResult);

$this->trackLog("addResult", $addResult);


if($statusResult && $addResult){

    //成功提交事务

    $db->commit();

    $this->trackLog("Trans", "事务处理成功， 订单状态和用户金额已经更新");

}else{

    //失败回滚

    $db->rollback();

    $this->trackLog("Trans", "事务处理失败， 已经回滚！ ");

}

}

}

//请不要修改或删除， 向支付宝返回成功信息

echo "success";

}
```

具体的操作请看钱包的接口文档。