支付宝当面付2.0 集成说明文档——PHP版

一．程序功能说明： 2

二．运行环境要求： 2

三．封装代码结构 2

四．接口方法 3

五．相应接口流程 4

1、条码支付流程： 4

2、查询接口流程 6

3、退款接口流程： 8

4、扫码支付（不含异步通知） 10

# 一．程序功能说明：

在支付宝标准SDK的基础上再做了一层封装， 即PHP版本demo中f2fpay文件夹(集成了支付轮询撤销逻辑)

# 二．运行环境要求：

**PHP5.5以上**

# 三．封装代码结构

**###f2fpay文件夹目录代码结构###**

|\_\_\_\_config

| |\_\_\_\_config.php //配置文件（appid、密钥等）

|\_\_\_\_log //日志文件夹

| |\_\_\_\_log.txt

|\_\_\_\_model

| |\_\_\_\_builder

| | |\_\_\_\_AlipayTradePayContentBuilder.php //条码支付请求bizContent结构体

| | |\_\_\_\_AlipayTradePrecreateContentBuilder.php //扫码支付(生成二维码)请求bizContent结构体

| | |\_\_\_\_AlipayTradeQueryContentBuilder.php //查询请求bizContent结构体

| | |\_\_\_\_AlipayTradeRefundContentBuilder.php //退款请求bizContent结构体

| | |\_\_\_\_AlipayTradeCancelContentBuilder.php //撤销请求bizContent结构体

| | |\_\_\_\_ExtendParams.php //扩展参数构造类

| | |\_\_\_\_GoodsDetail.php //商品详情参数构造类

| | |\_\_\_\_RoyaltyDetailInfo.php //分润参数构造类

| |\_\_\_\_result

| | |\_\_\_\_AlipayF2FPayResult.php //条码支付应答

| | |\_\_\_\_AlipayF2FPrecreateResult.php //扫码支付(生成二维码)应答

| | |\_\_\_\_AlipayF2FQueryResult.php //查询应答

| | |\_\_\_\_AlipayF2FRefundResult.php //退款应答

|\_\_\_\_service

| |\_\_\_\_AlipayTradeService.php //当面付2.0服务实现，包括条码支付（带轮询）、扫码支付、消费查询、消费退款

|\_\_\_\_barpay\_test.php //条码测试页面

|\_\_\_\_qrpay\_test.php //扫码（二维码）测试页面

|\_\_\_\_query\_test.php //查询测试页面

|\_\_\_\_refund\_test.php //退款测试页面

# 四．接口方法

**AlipayTradeService.php文件中相应方法：**

**// 当面付2.0条码支付(带轮询逻辑)**

**public function barPay($req)**

**// 当面付2.0消费查询**

**public function queryTradeResult($outTradeNo)**

**// 当面付2.0消费退款**

**public function refund($req)**

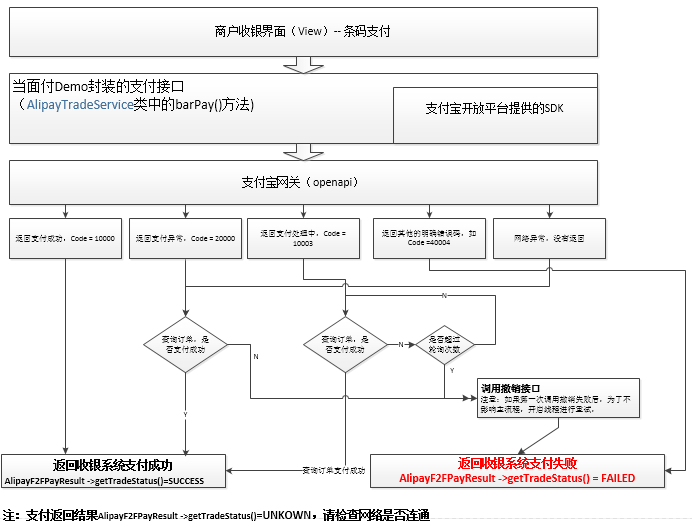
**//当面付2.0预下单(生成二维码,带轮询)**

**public function qrPay($req)**

# 五．相应接口流程

## 1、条码支付流程：

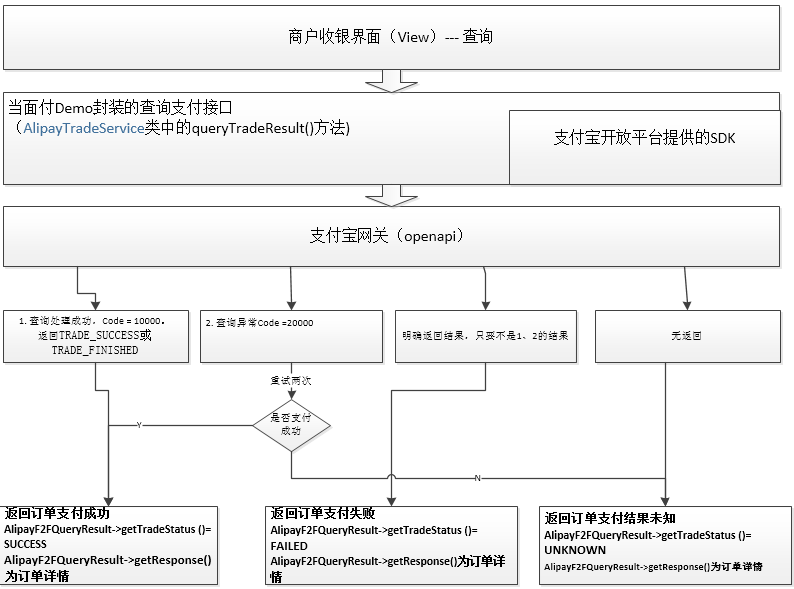
提供：支付、查询、退款 （条码支付），封装了支付宝SDK，提供给收银系统界面直接调用。



示例代码：

|  |
| --- |
| // 创建请求builder，设置请求参数 $barPayRequestBuilder = **new** AlipayTradePayContentBuilder(); $barPayRequestBuilder->setOutTradeNo($outTradeNo); $barPayRequestBuilder->setTotalAmount($totalAmount); $barPayRequestBuilder->setAuthCode($authCode);  // 调用barPay方法获取当面付应答 $barPay = **new** AlipayTradeService($config); $barPayResult = $barPay->barPay($barPayRequestBuilder);  //获得支付结果 **switch** ($barPayResult->getTradeStatus()) {  **case** "SUCCESS":  **echo** "支付宝支付成功:" . "<br>--------------------------<br>";  print\_r($barPayResult->getResponse());  **break**;  **case** "FAILED":  **echo** "支付宝支付失败!!!" . "<br>--------------------------<br>";  **if** (!**empty**($barPayResult->getResponse())) {  print\_r($barPayResult->getResponse());  }  **break**;  **case** "UNKNOWN":  **echo** "系统异常，订单状态未知!!!" . "<br>--------------------------<br>";  **if** (!**empty**($barPayResult->getResponse())) {  print\_r($barPayResult->getResponse());  }  **break**;  **default**:  **echo** "不支持的交易状态，交易返回异常!!!";  **break**; } |

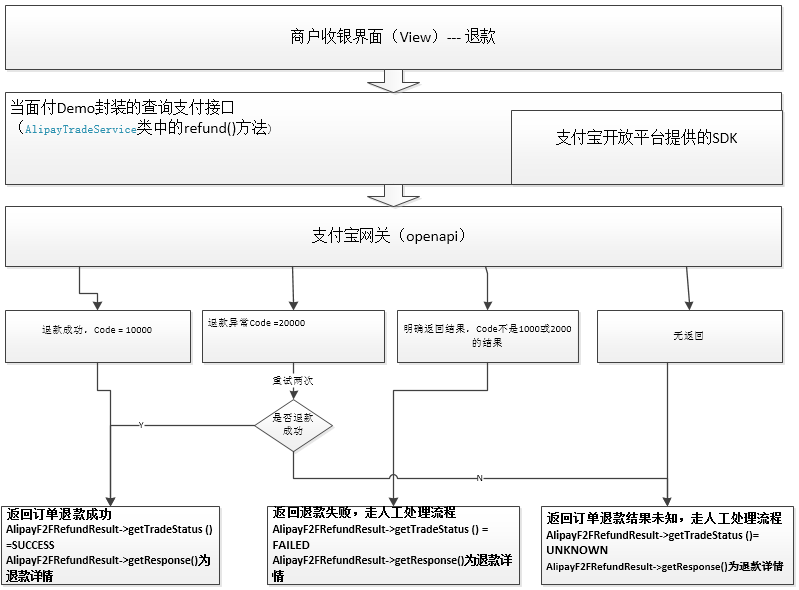
## 2、查询接口流程



示例代码

|  |
| --- |
| //获取商户订单号 $out\_trade\_no = trim($\_POST['out\_trade\_no']);  //初始化类对象，调用queryTradeResult方法获取查询应答  $queryResponse = **new** AlipayTradeService($config); $queryResult = $queryResponse->queryTradeResult($out\_trade\_no);  //根据查询返回结果状态进行业务处理 **switch** ($queryResult->getTradeStatus()){  **case** "SUCCESS":  **echo** "支付宝查询交易成功:"."<br>--------------------------<br>";  print\_r($queryResult->getResponse());  **break**;  **case** "FAILED":  **echo** "支付宝查询交易失败或者交易已关闭!!!"."<br>--------------------------<br>";  **if**(!**empty**($queryResult->getResponse())){  print\_r($queryResult->getResponse());  }  **break**;  **case** "UNKNOWN":  **echo** "系统异常，订单状态未知!!!"."<br>--------------------------<br>";  **if**(!**empty**($queryResult->getResponse())){  print\_r($queryResult->getResponse());  }  **break**;  **default**:  **echo** "不支持的查询状态，交易返回异常!!!";  **break**; } |

## 3、退款接口流程：

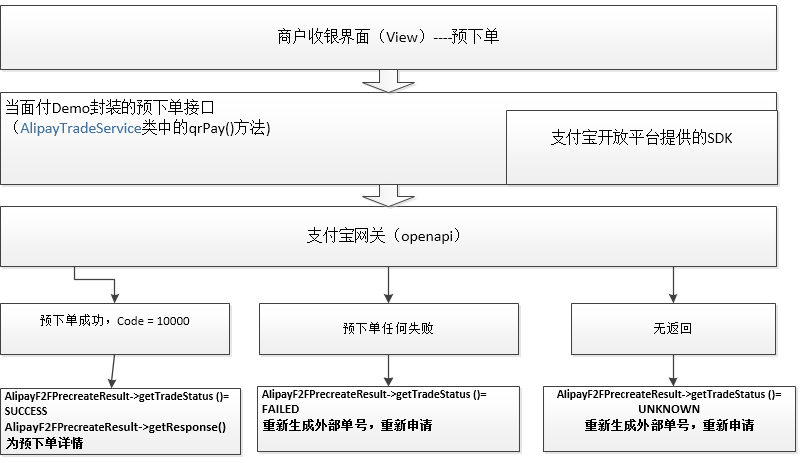


示例代码

|  |
| --- |
| //创建退款请求builder,设置参数 $refundRequestBuilder = **new** AlipayTradeRefundContentBuilder();  $refundRequestBuilder->setOutTradeNo($out\_trade\_no);  $refundRequestBuilder->setRefundAmount($refund\_amount);  $refundRequestBuilder->setOutRequestNo($out\_request\_no);  //初始化类对象,调用refund获取退款应答 $refundResponse = **new** AlipayTradeService($config); $refundResult = $refundResponse->refund($refundRequestBuilder);  //根据交易状态进行处理 **switch** ($refundResult->getTradeStatus()){  **case** "SUCCESS":  **echo** "支付宝退款成功:"."<br>--------------------------<br>";  print\_r($refundResult->getResponse());  **break**;  **case** "FAILED":  **echo** "支付宝退款失败!!!"."<br>--------------------------<br>";  **if**(!**empty**($refundResult->getResponse())){  print\_r($refundResult->getResponse());  }  **break**;  **case** "UNKNOWN":  **echo** "系统异常，订单状态未知!!!"."<br>--------------------------<br>";  **if**(!**empty**($refundResult->getResponse())){  print\_r($refundResult->getResponse());  }  **break**;  **default**:  **echo** "不支持的交易状态，交易返回异常!!!";  **break**; } |

## 4、扫码支付（不含异步通知）

预下单（扫码支付）接口



示例代码：

|  |
| --- |
| // 创建请求builder，设置请求参数 $qrPayRequestBuilder = **new** AlipayTradePrecreateContentBuilder(); $qrPayRequestBuilder->setOutTradeNo($outTradeNo); $qrPayRequestBuilder->setTotalAmount($totalAmount);  // 调用qrPay方法获取当面付应答 $qrPay = **new** AlipayTradeService($config); $qrPayResult = $qrPay->qrPay($qrPayRequestBuilder);  // 根据状态值进行业务处理 **switch** ($qrPayResult->getTradeStatus()){  **case** "SUCCESS":  **echo** "支付宝支付成功:"."<br>--------------------------<br>";  print\_r($qrPayResult->getResponse());  **break**;  **case** "FAILED":  **echo** "支付宝支付失败!!!"."<br>--------------------------<br>";  **if**(!**empty**($qrPayResult->getResponse())){  print\_r($qrPayResult->getResponse());  }  **break**;  **case** "UNKNOWN":  **echo** "系统异常，订单状态未知!!!"."<br>--------------------------<br>";  **if**(!**empty**($qrPayResult->getResponse())){  print\_r($qrPayResult->getResponse());  }  **break**;  **default**:  **echo** "不支持的交易状态，交易返回异常!!!";  **break**; } |

**扫码支付获得支付结果：**

方案一： 采用上面的查询接口。

方案二： 采用轮询逻辑：



**扫码支付获得支付结果：**

方案一： 采用上面的查询接口。

方案二： 采用轮询逻辑：



# 

