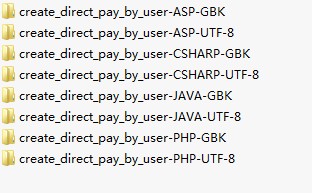
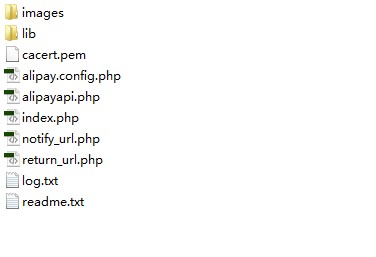
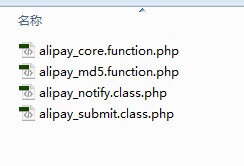
二、重新整理接口包文件，这一步应该算是比较关键的（个人认为）  
下载下来的接口包文件有很多语言的源码，  
[](http://www.thinkphp.cn/Uploads/editor/2013-07-16/51e444726490b.jpg)  
我们选择 create\_direct\_pay\_by\_user-PHP-UTF-8 这个名称的接口文件。里面包括如下文件：  
[](http://www.thinkphp.cn/Uploads/editor/2013-07-16/51e444ac637ec.jpg)  
images文件里是支付宝相关的一些标志的图片，我们暂不管他，lib文件很重要，是整个接口的核心类文件；  
alipay.config.php是相关参数的配置文件  
alipayapi.php 是支付宝接口入口文件  
notify\_url.php 是服务器异步通知页面文件；  
return\_url.php 是页面跳转同步通知文件；  
  
在ThinkPHP的框架文件下，找到Extend 进入，再进入Vendor,在Vendor文件夹下，新建文件夹Alipay，把支付宝作为第三方类库引入。然后，复制支付宝接口文件包中lib文件里的所有文件。一共4个文件，如下：  
[](http://www.thinkphp.cn/Uploads/editor/2013-07-16/51e4448a25728.jpg)  
现在对以上文件进行重命名，  
alipay\_core.function.php重命名为：Corefunction.php;  
alipay\_md5.function.php重命名为：Md5function.php；  
alipay\_notify.class.php重命名为：Notify.php；  
alipay\_submit.class.php重命名为：Submit.php；  
  
然后，打开Submit.php文件，把以下代码去掉；

1. require\_once("alipay\_core.function.php");
2. require\_once("alipay\_md5.function.php");

复制代码

同样，打开Notify.php文件，把以下两段代码去掉

1. require\_once("alipay\_core.function.php");
2. require\_once("alipay\_md5.function.php");

复制代码

**为什么要去掉以上两个文件中的这两段代码，因为在项目中调用接口文件的时候，我把所有4个核心文件都通过vendor来进行引入。所以，这不再需要导入。**  
  
到此，支付宝接口包相关核心类库的整理基本完成。现在开始在项目中调用；  
  
三、在项目中调用支付宝接口  
调用分两步：  
1、在配置文件中Conf/Config.php文件中对支付宝相关参数进行配置：

//支付宝配置参数

'alipay\_config'=>**array**(

'partner' =>'2088521139824685', //这里是你在成功申请支付宝接口后获取到的PID；

'key'=>'5e1l9eo803fuftaom0z2cadis8mr1jig',//这里是你在成功申请支付宝接口后获取到的Key

'sign\_type'=>strtoupper('MD5'),

'input\_charset'=> strtolower('utf-8'),

'cacert'=> getcwd().'\\cacert.pem',

'transport'=> 'http',

),

//以上配置项，是从接口包中alipay.config.php 文件中复制过来，进行配置；

'alipay' =>**array**(

//这里是卖家的支付宝账号，也就是你申请接口时注册的支付宝账号

'seller\_email'=>'zxn@hzylwl.com',

//这里是异步通知页面url，提交到项目的Pay控制器的notifyurl方法；

'notify\_url'=>'http://127.0.0.1/mypro/Myresp/index.php/Home/Pay/notifyurl',

//这里是页面跳转通知url，提交到项目的Pay控制器的returnurl方法；

'return\_url'=>'http://127.0.0.1/mypro/Myresp/index.php/Home/Pay/returnurl',

//支付成功跳转到的页面，我这里跳转到项目的User控制器，myorder方法，并传参payed（已支付列表）

'successpage'=>'Pay/returnurl?ordtype=payed',

//支付失败跳转到的页面，我这里跳转到项目的User控制器，myorder方法，并传参unpay（未支付列表）

'errorpage'=>'Pay/returnurl?ordtype=unpay',

),复制代码

2、新建一个PayAction控制器代码如下：

<?php

namespace Home\Controller;

use Think\Controller;

use Home\Model\UserModel;

use Home\Model\OrderlistModel;

use Home\Model\PayorderModel;

class PayController extends Controller {

//在类初始化方法中，引入相关类库

public function \_initialize() {

vendor('Alipay.Corefunction');

vendor('Alipay.Md5function');

vendor('Alipay.Notify');

vendor('Alipay.Submit');

}

//doalipay方法

/\*该方法其实就是将接口文件包下alipayapi.php的内容复制过来

然后进行相关处理

\*/

public function doalipay(){

$id=$\_SESSION['id'];

//接收到数据

$res = M('user');

$user\_data = $res->find($id);

$money = isset($\_POST['money'])?$\_POST['money']:"";

$name=isset($\_POST['order\_payname'])?$\_POST['order\_payname']:"";

$phone=isset($\_POST['order\_phone'])?$\_POST['order\_phone']:"";

if(empty($name) || empty($phone)){

$this->error('打款人和电话不能为空',\_\_MODULE\_\_."/Order/listmsg");exit;

}

$time=date("y-m-d H:i:s");

$a=mt\_rand(100000000, 999999999);

$order\_no='1479'.$a;

switch ($\_POST['order\_platform']){

case '1': $pname='快用'; break;

case '2': $pname='当乐'; break;

case '3': $pname='夜神'; break;

case '4': $pname='TT语音'; break;

case '5': $pname='乐8'; break;

case '6': $pname='虫虫'; break;

case '7': $pname='安峰'; break;

case '8': $pname='机峰'; break;

}

//将数据保存到数据库

$data=array('user\_id'=>$id,

'order\_no'=>$order\_no,

'order\_payname'=>$name,

'order\_payphone'=>$phone,

'order\_money'=>$money,

'order\_recharge'=>$\_POST['money'],

'order\_adminNo'=>$\_POST['order\_adminNo'],

'order\_channel'=>$\_POST['order\_channel'],

'order\_platform'=>$pname,

'order\_gamename'=>$\_POST['order\_gamename'],

'order\_time'=>$time,

);

/\*echo "<pre>";

print\_r($data);

exit;\*/

/\*

$db = M('payorder');

$info = $db->add($data);

if($info){

//支付成功

if($money>$user\_data['user\_money']){

$this->error("您的余额不足，请充值！");

}else if($money==0 || empty($money)){

$this->error("请输入您要充值的金额！");

}else{

$num=$user\_data['user\_money']-$money;//用户金额-充值金额

$res->where('user\_id='.$id)->setField('user\_money',$num);

$this->success("充值成功");

}

}else{

$this->error('支付失败，请再试一次');

}

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

把alipayapi.php中复制过来的如下两段代码去掉，

第一段是引入配置项，

第二段是引入submit.class.php这个类。

为什么要去掉？？

第一，配置项的内容已经在项目的Config.php文件中进行了配置，我们只需用C函数进行调用即可；

第二，这里调用的submit.class.php类库我们已经在PayAction的\_initialize()中已经引入；所以这里不再需要；

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// require\_once("alipay.config.php");

// require\_once("lib/alipay\_submit.class.php");

//生成订单号，当前的时间戳加一个六位的随机数，用md5加密

$no=md5(time().(mt\_rand(100000,999999)));

//生成订单名

$sub=$pname.$name;

//echo $sub;exit;

//这里我们通过TP的C函数把配置项参数读出，赋给$alipay\_config；

$alipay\_config=C('alipay\_config');

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*请求参数\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

$payment\_type = "1"; //支付类型 //必填，不能修改

$notify\_url = C('alipay.notify\_url'); //服务器异步通知页面路径

$return\_url = C('alipay.return\_url'); //页面跳转同步通知页面路径

$seller\_email = C('alipay.seller\_email');//卖家支付宝帐户必填

//$out\_trade\_no = $\_POST['trade\_no'];//商户订单号 通过支付页面的表单进行传递，注意要唯一！

$out\_trade\_no =$no;

//$subject = $\_POST['ordsubject']; //订单名称 //必填 通过支付页面的表单进行传递

$subject = $sub;

//$total\_fee = $\_POST['ordtotal\_fee']; //付款金额 //必填 通过支付页面的表单进行传递

$total\_fee = $money;

//$body = $\_POST['ordbody']; //订单描述 通过支付页面的表单进行传递

$body = $\_POST['order\_remark'];

// $show\_url = $\_POST['ordshow\_url']; //商品展示地址 通过支付页面的表单进行传递

$anti\_phishing\_key = "";//防钓鱼时间戳 //若要使用请调用类文件submit中的query\_timestamp函数

$exter\_invoke\_ip = get\_client\_ip(); //客户端的IP地址

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//构造要请求的参数数组，无需改动

$parameter = array(

"service" => "create\_direct\_pay\_by\_user",

"partner" => trim($alipay\_config['partner']),

"payment\_type" => $payment\_type,

"notify\_url" => $notify\_url,

"return\_url" => $return\_url,

"seller\_email" => $seller\_email,

"out\_trade\_no" => $out\_trade\_no,

"subject" => $subject,

"total\_fee" => $total\_fee,

"body" => $body,

"show\_url" => '',

"anti\_phishing\_key" => $anti\_phishing\_key,

"exter\_invoke\_ip" => $exter\_invoke\_ip,

"\_input\_charset" => trim(strtolower($alipay\_config['input\_charset']))

);

//建立请求

$alipaySubmit = new \ AlipaySubmit($alipay\_config);

$html\_text = $alipaySubmit->buildRequestForm($parameter,"post", "确认");

echo $html\_text;

$db = M('payorder');

$db->add($data);

$data=array('user\_id'=>$id,'user\_name'=>$name,'ordid'=>$no,'ordfee'=>$money);

$orderlist=new OrderlistModel();

$orderlist->add($data);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

服务器异步通知页面方法

其实这里就是将notify\_url.php文件中的代码复制过来进行处理

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

function notifyurl(){

/\*

同理去掉以下两句代码；

\*/

//require\_once("alipay.config.php");

//require\_once("lib/alipay\_notify.class.php");

//这里还是通过C函数来读取配置项，赋值给$alipay\_config

$alipay\_config=C('alipay\_config');

//计算得出通知验证结果

$alipayNotify = new \ AlipayNotify($alipay\_config);

$verify\_result = $alipayNotify->verifyNotify();

if($verify\_result) {

//验证成功

//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

$out\_trade\_no = $\_POST['out\_trade\_no']; //商户订单号

$trade\_no = $\_POST['trade\_no']; //支付宝交易号

$trade\_status = $\_POST['trade\_status']; //交易状态

$total\_fee = $\_POST['total\_fee']; //交易金额

$notify\_id = $\_POST['notify\_id']; //通知校验ID。

$notify\_time = $\_POST['notify\_time']; //通知的发送时间。格式为yyyy-MM-dd HH:mm:ss。

$buyer\_email = $\_POST['buyer\_email']; //买家支付宝帐号；

$parameter = array(

"out\_trade\_no" => $out\_trade\_no, //商户订单编号；

"trade\_no" => $trade\_no, //支付宝交易号；

"total\_fee" => $total\_fee, //交易金额；

"trade\_status" => $trade\_status, //交易状态

"notify\_id" => $notify\_id, //通知校验ID。

"notify\_time" => $notify\_time, //通知的发送时间。

"buyer\_email" => $buyer\_email, //买家支付宝帐号；

);

if($\_POST['trade\_status'] == 'TRADE\_FINISHED') {

//

}else if ($\_POST['trade\_status'] == 'TRADE\_SUCCESS') {if(!checkorderstatus($out\_trade\_no)){

orderhandle($parameter);

//进行订单处理，并传送从支付宝返回的参数；

}

}

echo "success"; //请不要修改或删除

}else {

//验证失败

echo "fail";

}

}

/\*

页面跳转处理方法；

这里其实就是将return\_url.php这个文件中的代码复制过来，进行处理；

\*/

function returnurl(){

//头部的处理跟上面两个方法一样，这里不罗嗦了！

$alipay\_config=C('alipay\_config');

$alipayNotify = new \ AlipayNotify($alipay\_config);//计算得出通知验证结果

$verify\_result = $alipayNotify->verifyReturn();

if($verify\_result) {

//验证成功

//获取支付宝的通知返回参数，可参考技术文档中页面跳转同步通知参数列表

$out\_trade\_no = $\_GET['out\_trade\_no']; //商户订单号

$trade\_no = $\_GET['trade\_no']; //支付宝交易号

$trade\_status = $\_GET['trade\_status']; //交易状态

$total\_fee = $\_GET['total\_fee']; //交易金额

$notify\_id = $\_GET['notify\_id']; //通知校验ID。

$notify\_time = $\_GET['notify\_time']; //通知的发送时间。

$buyer\_email = $\_GET['buyer\_email']; //买家支付宝帐号；

$parameter = array(

"out\_trade\_no" => $out\_trade\_no, //商户订单编号；

"trade\_no" => $trade\_no, //支付宝交易号；

"total\_fee" => $total\_fee, //交易金额；

"trade\_status" => $trade\_status, //交易状态

"notify\_id" => $notify\_id, //通知校验ID。

"notify\_time" => $notify\_time, //通知的发送时间。

"buyer\_email" => $buyer\_email, //买家支付宝帐号

);

if($\_GET['trade\_status'] == 'TRADE\_FINISHED' || $\_GET['trade\_status'] == 'TRADE\_SUCCESS') {

if(!checkorderstatus($out\_trade\_no)){

orderhandle($parameter); //进行订单处理，并传送从支付宝返回的参数；

}

$this->redirect(C('alipay.successpage'));//跳转到配置项中配置的支付成功页面；

}else {

echo "trade\_status=".$\_GET['trade\_status'];

$this->redirect(C('alipay.errorpage'));//跳转到配置项中配置的支付失败页面；

}

}else {

//验证失败

//如要调试，请看alipay\_notify.php页面的verifyReturn函数

$this->success('支付成功',\_\_MODULE\_\_."/Order/listmsg");

}

}

}

?>

复制代码

3、这里有几个支付处理过程中需要用到的函数，我把这些函数写到了项目的Common/common.php中，这样不用手动调用，即可直接使用这些函数，代码如下：

//////////////////////////////////////////////////////

//Orderlist数据表，用于保存用户的购买订单记录；

/\* Orderlist数据表结构；

CREATE TABLE `tb\_orderlist` (

  `id` int(11) NOT NULL AUTO\_INCREMENT,

  `userid` int(11) DEFAULT NULL,购买者userid

  `username` varchar(255) DEFAULT NULL,购买者姓名

  `ordid` varchar(255) DEFAULT NULL,订单号

  `ordtime` int(11) DEFAULT NULL,订单时间

  `productid` int(11) DEFAULT NULL,产品ID

  `ordtitle` varchar(255) DEFAULT NULL,订单标题

  `ordbuynum` int(11) DEFAULT '0',购买数量

  `ordprice` float(10,2) DEFAULT '0.00',产品单价

  `ordfee` float(10,2) DEFAULT '0.00',订单总金额

  `ordstatus` int(11) DEFAULT '0',订单状态

  `payment\_type` varchar(255) DEFAULT NULL,支付类型

  `payment\_trade\_no` varchar(255) DEFAULT NULL,支付接口交易号

  `payment\_trade\_status` varchar(255) DEFAULT NULL,支付接口返回的交易状态

  `payment\_notify\_id` varchar(255) DEFAULT NULL,

  `payment\_notify\_time` varchar(255) DEFAULT NULL,

  `payment\_buyer\_email` varchar(255) DEFAULT NULL,

  `ordcode` varchar(255) DEFAULT NULL,       //这个字段不需要的，大家看我西面的修正补充部分的说明！

  `isused` int(11) DEFAULT '0',

  `usetime` int(11) DEFAULT NULL,

  `checkuser` int(11) DEFAULT NULL,

  PRIMARY KEY (`id`)

) ENGINE=MyISAM AUTO\_INCREMENT=5 DEFAULT CHARSET=utf8;

\*/

//在线交易订单支付处理函数

//函数功能：根据支付接口传回的数据判断该订单是否已经支付成功；

//返回值：如果订单已经成功支付，返回true，否则返回false；

**function** checkorderstatus($ordid){

$Ord=M('Orderlist');

// echo $Ord->getLastSql(); echo 1;exit;

$ordstatus=$Ord->where("ordid='$ordid'")->getField('ordstatus');

**if**($ordstatus==1){

**return true**;

}**else**{

**return false**;

}

}

//处理订单函数

//更新订单状态，写入订单支付后返回的数据

**function** orderhandle($parameter){

$ordid=$parameter['out\_trade\_no'];

$data['payment\_trade\_no'] =$parameter['trade\_no'];

$data['payment\_trade\_status'] =$parameter['trade\_status'];

$data['payment\_notify\_id'] =$parameter['notify\_id'];

$data['payment\_notify\_time'] =$parameter['notify\_time'];

$data['payment\_buyer\_email'] =$parameter['buyer\_email'];

$data['ordstatus'] =1;

$Ord=M('Orderlist');

// echo $Ord->getLastSql(); echo 2;exit;

$Ord->where("ordid='$ordid'")->save($data);

1. }
2. **/\*-----------------------------------**
3. **2013.8.13更正**
4. **下面这个函数，其实不需要，大家可以把他删掉，**
5. **具体看我下面的修正补充部分的说明**
6. **------------------------------------\*/**
7. //获取一个随机且唯一的订单号；
8. function getordcode(){
9. $Ord=M('Orderlist');
10. $numbers = range (10,99);
11. shuffle ($numbers);
12. $code=array\_slice($numbers,0,4);
13. $ordcode=$code[0].$code[1].$code[2].$code[3];
14. $oldcode=$Ord->where("ordcode='".$ordcode."'")->getField('ordcode');
15. if($oldcode){
16. getordcode();
17. }else{
18. return $ordcode;
19. }
20. }

复制代码

四、总结几点  
1、接口包中lib文件中的文件复制到Vendor后，重命名为TP规范的命名规则，为的是调用方便，当然你要改成其他名称也可以；  
2、把执行支付操作（doalipay），处理异步返回结果（notifyurl），处理跳转返回结果（returnurl）三个支付接口的核心页面写到一个PayAction控制器中。  
3、提交支付的页面中，可以在提交之前先把一些参数要传递的内容先通过隐藏域的方法组合好，比如金额先计算好，订单名称，订单描述等先用字符串组合好。然后提交表单，这样，在doalipay方法中只要直接构造传递参数，直接进行提交就行过了。  
4、支付返回后的处理因为要在异步和跳转两个方法中都要进行相应的判断和处理，所以，把这些判断和处理写到一个自定义函数中，这样只要调用函数即可，使得代码更加清晰明了。  
5、notify\_url和return\_url两种模式的返回url必须使用http://xxxxxxx这样的绝对路径，因为里是从支付宝平台返回到你的项目页面。不能使用相对路径。  
  
以上代码在ThinkPHP3.0中正常使用！！  
  
**------------------------修正补充！！2013.08.13------------------------------**  
在第三部分中Orderlist数据表结构中，我有一个字段是OrdCode，这个字段是我系统中用来发送短信给客户的消费密码，也就是客户凭手机短信来消费时就要验证这个字段。  
其实，大家在做系统的时候，可以把这个字段忽略，可以不用他。代码最后部分中，有一个获取一个随机且唯一的订单号的函数 getordcode()，这里我其实写错了，不是获取订单号，是ordcode，也就是消费密码，这个函数也不需要。系统中的订单号（ordid字段），我用的是时间戳。  
在此修正！  
  
**--------------------解决签名错误问题 修正 13-08-16------------------------**  
有人说在在调试时，签名出现无法通过的问题，产生问题的原因是在返回的URL地址中返回的参数中，可能存在\_\_URL\_\_这样的字符串。导致无法正确过滤参数。  
  
**解决办法:**  
方法1：  
在向支付宝提交需要的参数时，不要使用\_\_URL\_\_,\_\_PUBLIC\_\_等TP中的模版替换变量，如果TP对这些变量解析不成功，会直接传递过去，所以，在这些地方直接使用原始的URL地址。  
  
方法2:  
在接口的Core文件中，加入改造后的过滤函数，如下：

1. /\*\*
2. \* 除去数组中的空值和签名参数
3. \* @param $para 签名参数组
4. \* return 去掉空值与签名参数后的新签名参数组
5. \*/
6. function paraFilter($para) {
7. $para\_filter = array();
8. while (list ($key, $val) = each ($para)) {
9. if($key == "sign" || $key == "sign\_type" || $key == '\_URL\_' || $val == "")continue;    //添加了$key == '\_URL\_'
10. else    $para\_filter[$key] = $para[$key];
11. }
12. return $para\_filter;
13. }
14. function myparaFilter($para) {
15. $para\_filter = array();
16. while (list ($key, $val) = each ($para)) {
17. if($key == '\_URL\_')continue;
18. else $para\_filter[$key] = $para[$key];
19. }
20. return $para\_filter;
21. }

复制代码

**=============================================**