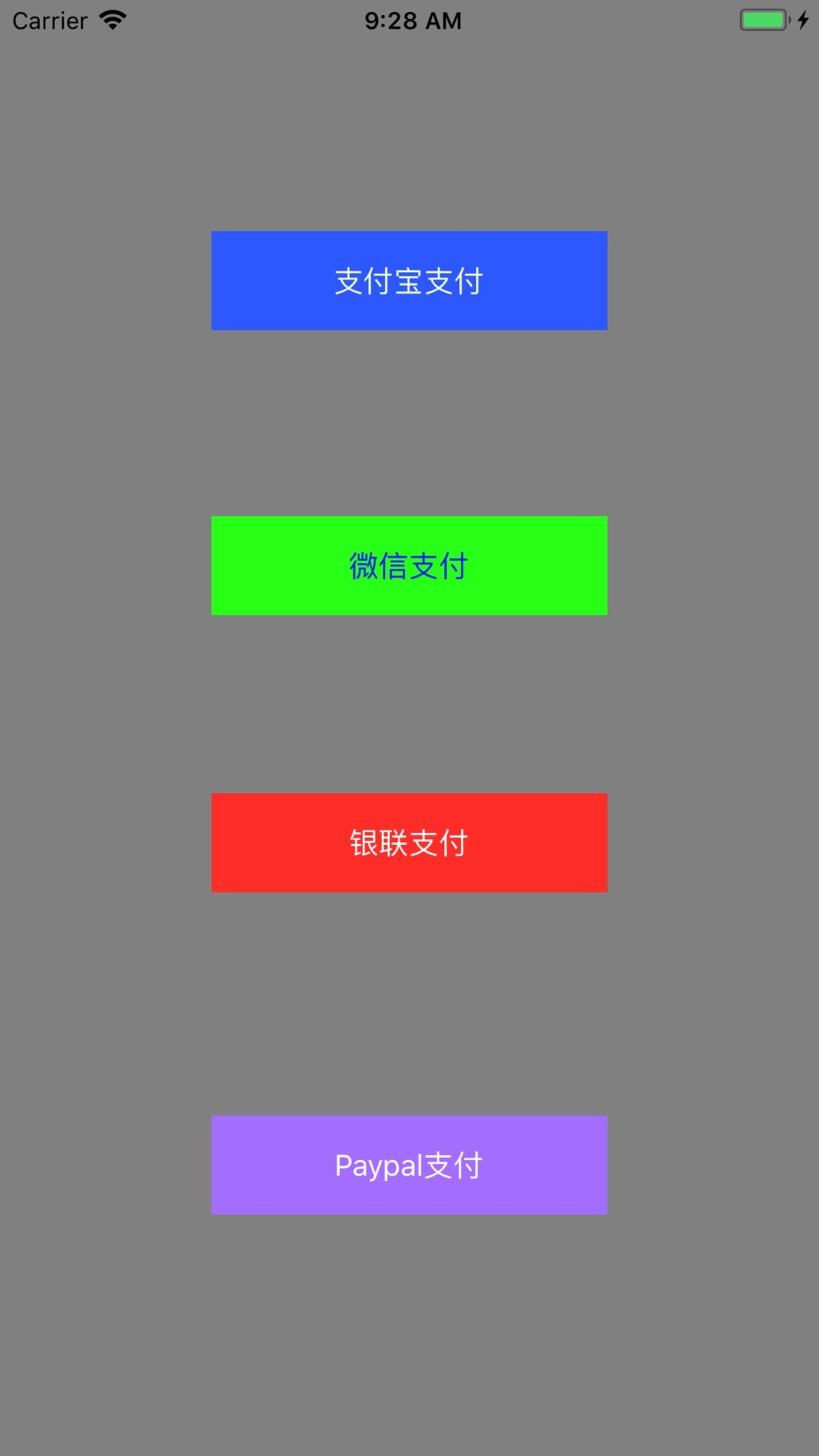
**iOS App支付指南**

具体参考我的demo



**APP支付是指商户通过在移动端应用APP中集成开放SDK调起移动支付模块来完成支付，适用于在移动端APP中集成移动支付功能的场景。**

## 微信支付SDK接入指南: <https://open.weixin.qq.com/cgi-bin/showdocument?action=dir_list&t=resource/res_list&verify=1&id=1417694084&token=&lang=zh_CN>

## 支付宝SDK接入指南: <https://doc.open.alipay.com/docs/doc.htm?spm=a219a.7629140.0.0.d4oVLq&treeId=204&articleId=105295&docType=1>

## 银联支付控件SDK接入指南: <https://open.unionpay.com/ajweb/help/file/techFile?productId=3>

## Paypal SDK 接入指南: <https://developer.paypal.com/webapps/developer/docs/integration/mobile/ios-integration-guide/>

适用于商户在移动端APP中集成支付宝/微信/银联/Paypal支付功能。商户APP调用提供的SDK调用支付模块，商户APP会跳转到支付宝/微信中完成支付，支付完后跳回到商户APP内，最后展示支付结果。目前移动支付支持手机系统有：iOS（苹果）、Android（安卓）

**交互细节如下:**

**步骤1：**用户进入商户APP，选择商品下单、确认购买，进入支付环节。商户服务后台生成支付订单，签名后将数据传输到APP端

**步骤2：**用户点击后发起支付操作，进入到微信界面，调起微信支付，出现确认支付界面。

**步骤3：**用户确认收款方和金额，点击立即支付后出现输入密码界面，可选择零钱或银行卡支付。

**步骤4：**输入正确密码后，支付完成，用户端微信出现支付详情页面。

**步骤5：**：回跳到商户APP中，商户APP根据支付结果个性化展示订单处理结果。

**业务流程**

以下是交互时序图，统一下单API、支付结果通知API和查询订单API等都涉及签名过程，调用都必须在商户服务器端完成。

商户系统和微信支付系统主要交互说明：

步骤1：用户在商户APP中选择商品，提交订单，选择微信支付。

步骤2：商户后台收到用户支付单，调用微信支付统一下单接口。

步骤3：统一下单接口返回正常的prepay\_id，再按签名规范重新生成签名后，将数据传输给APP。参与签名的字段名为appid，partnerid，prepayid，noncestr，timestamp，package。注意：package的值格式为Sign=WXPay

步骤4：商户APP调起微信支付。

步骤5：商户后台接收支付通知。

步骤6：商户后台查询支付结果。

在智能手机上使用App应用支付的idea是伴随着以iPhone为代表的现代手机而出现，得益于iPhone开启的App使用习惯，将其运用于支付就显得非常自然。其工作机制非常简单，通过扫描二维码（Barcode）的形式来对商品完成支付（如上图的星巴克Android App支付流程）。尽管这看起来不比常用的现金或者信用卡支付明显方便，但是与手机的结合，使得整个购物流程更加的移动化、智能化，而不仅仅是支付形式的便利。

一些创业公司已经在试图使用该技术来建立自己的支付网络，但是更多的则是被类似星巴克这样的有具体业务的公司使用。其基本原理显而易见：对于个体公司来说，App应用支付不仅能提升客户对其优惠卡的使用，也为其拓宽了促销的市场渠道。 而对于消费者来说，通过手机的应用进行支付不仅是件非常自然的事情，同时还能从中获得优惠折扣，何乐而不为呢

APP支付服务端详解

相比微信支付，支付宝APP支付提供了支付分装类，下面将实现支付宝APP支付、订单查询、支付结果异步通知、APP支付申请参数说明，以及服务端返回APP端发起支付的签名、商户私钥、支付宝公钥的配置使用等。

支付注意事项

1、APP支付不能在沙箱测试、只能申请上线测试   
2、需要创建RSA密钥[设置文档](https://doc.open.alipay.com/docs/doc.htm?spm=a219a.7629140.0.0.c851tN&treeId=193&articleId=105310&docType=1)，设置后上传rsa\_public\_key.pem【开发者公钥，上传时需要去掉公钥的头和尾】上传成功后换取支付宝公钥，为项目的alipay\_public\_key.pem   
3、rsa\_private\_key\_pkcs8.pem【开发者私钥】，去掉头和尾为项目的alipay\_private\_key\_pkcs8.pem   
4、需要导入所需支付包：alipay-sdk-java.jar 和 commons-logging.jar，具体参考：[服务端SDK](https://doc.open.alipay.com/docs/doc.htm?spm=a219a.7629140.0.0.nDV903&treeId=54&articleId=103419&docType=1)

支付流程

支付文档参考：[支付文档](https://doc.open.alipay.com/doc2/apiList?docType=4)，[支付文档2](https://doc.open.alipay.com/doc2/detail.htm?spm=a219a.7629140.0.0.x7kkCI&treeId=204&articleId=105465&docType=1)

**APP支付**：服务器端按照文档【统一收单交易支付接口】创建支付OrderStr返回APP端——-APP端拿到OrderStr发起支付—–支付宝服务器端回调服务端异步通知接口——-服务器端按照【App支付结果异步通知】校验签名等做业务逻辑处理

**APP支付订单查询**：服务器端调用【统一收单线下交易查询】查询支付订单

**APP支付申请退款**：每笔支付可以申请多次退款，但退款总金额不能超过支付金额，调用【统一收单交易退款接口】发起退款申请

**ＡPP支付退款查询**：服务端调用【 统一收单交易退款查询】查询退款订单信息

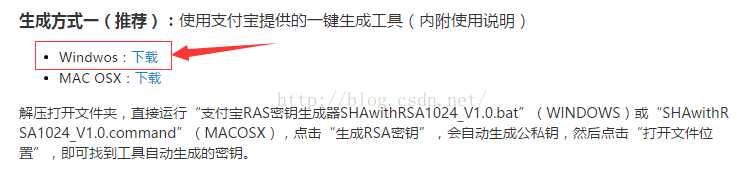
1. 首先，你的要有一个企业的账户，并且已经和支付宝平台签约

（具体操作查看<https://doc.open.alipay.com/doc2/detail.htm?treeId=58&articleId=103542&docType=1>）

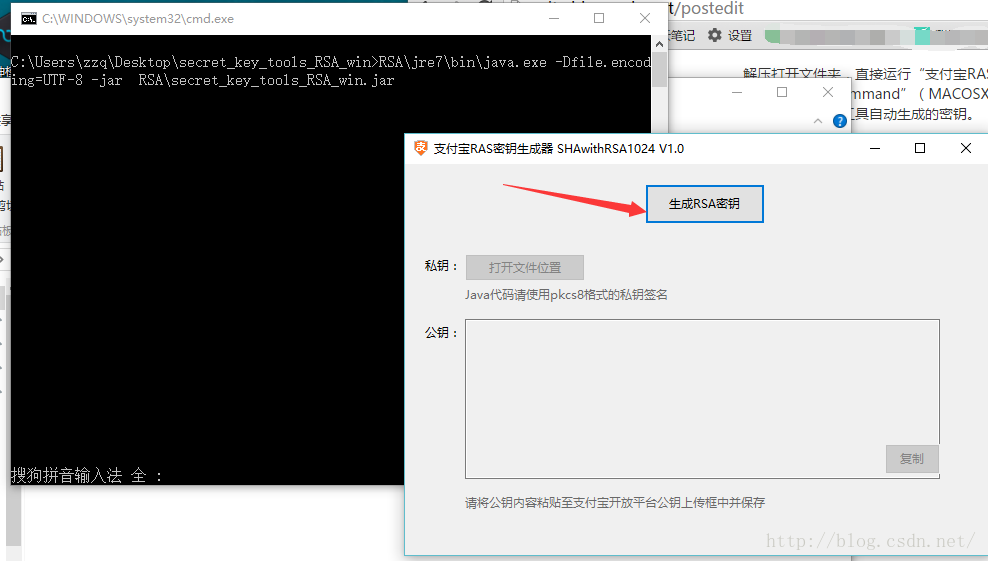
2然后就是配置RAS公钥和私钥

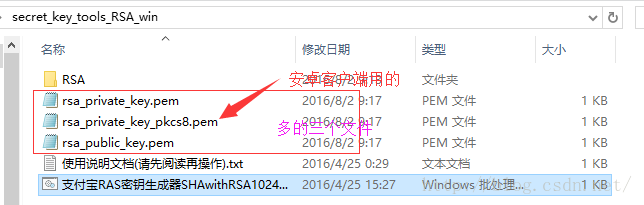
（文档 ：<https://doc.open.alipay.com/doc2/detail.htm?treeId=58&articleId=103242&docType=1>）

我们这里介绍的是使用方法一：因为简单....

首先我们将上面链接的

按照上面的步骤操作，能够看到

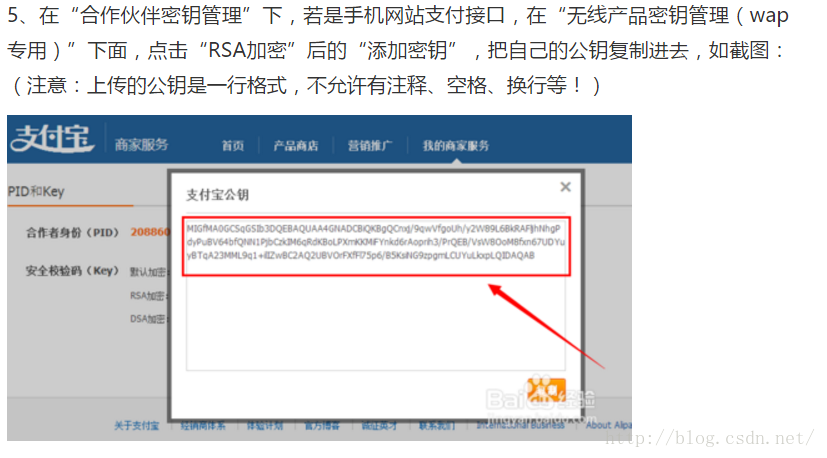


点击，可以看到和

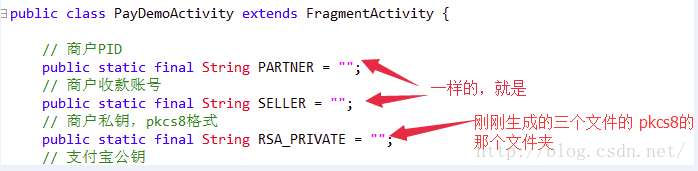


3将生成的公钥复制传到签约账号上：

<http://jingyan.baidu.com/article/9faa7231b702bb473c28cb03.html>



4.之后就是修改demo中的



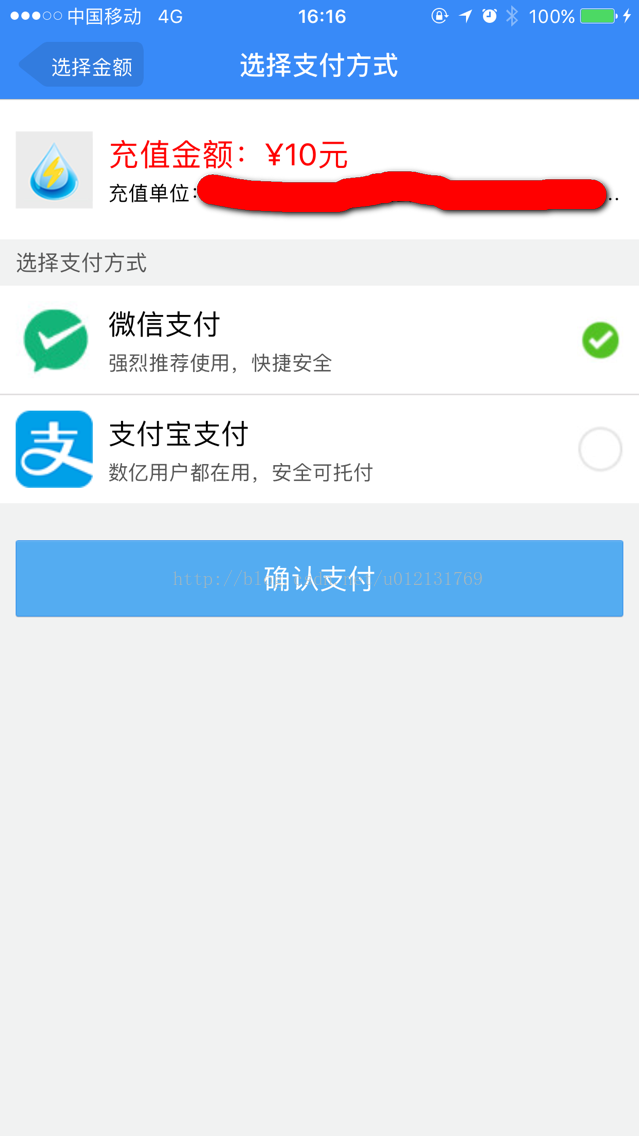
这个时候你就可以转账使用了。。。。。

因为demo平台上有，所以这里就不贴demo了

# ****支付宝APP支付（IOS手机端+java后台）版****

## 0.****介绍预览****

针对需要在IOS手机上接入原生微信支付场景，调用微信进行支付。如图：

## 1.****资料准备****

### ****1.1 账号注册****

打开https://openhome.alipay.com，注册支付宝开放平台开发者账号

### ****1.2 开发者认证****

登录，进入开发者中心，进行开发者资质认证，并创建移动应用。

### ****1.3 签约应用****

创建应用后上传相关资料，上线应用并通过审核。审核通过后点击应用，进行签约，此步骤不能省略，否则或报ISV权限不足。



### ****1.4 应用配置****

打开应用信息，配置一下内容，接口签名可下载支付宝签名验签工具进行生成，生成完成后将公钥复制到平台上。



## 2.****Java后台开发****

添加依赖

**[html]** [view plain](http://blog.csdn.net/u012131769/article/details/76639527) [copy](http://blog.csdn.net/u012131769/article/details/76639527)

[print?](http://blog.csdn.net/u012131769/article/details/76639527)

1. <!-- 支付宝支付依赖 -->
2. **<dependency>**
3. **<groupId>**com.pentahohub.nexus**</groupId>**
4. **<artifactId>**alipay-sdk-java**</artifactId>**
5. **<version>**20170615110434**</version>**
6. **<scope>**system**</scope>**
7. **<systemPath>**${basedir}/libs/alipay-sdk-java20170615110434.jar**</systemPath>**
8. **</dependency>**
9. **<dependency>**
10. **<groupId>**commons-logging**</groupId>**
11. **<artifactId>**commons-logging**</artifactId>**
12. **<version>**1.1.1**</version>**
13. **</dependency>**

生成统一订单

**[java]** [view plain](http://blog.csdn.net/u012131769/article/details/76639527) [copy](http://blog.csdn.net/u012131769/article/details/76639527)

[print?](http://blog.csdn.net/u012131769/article/details/76639527)

1. /\*\*
2. \* account 总金额
3. \* sid 用户id
4. \* desc 节点id
5. \* @param request
6. \* @return
7. \*/
8. @RequestMapping(value="/pay/alipay/signprams",produces="application/json;charset=utf-8")
9. @ResponseBody
10. **public** String signprams(HttpServletRequest request){
11. String res = "{code:404}";
12. **try**{
13. String totalAmount = request.getParameter("account");
14. String sid = request.getParameter("sid");
15. String nid = request.getParameter("desc");
16. String subject = "电费充值";
17. String body = "贵大电费预付费充值";
18. String prefix = "ALE"+sid+"E";
19. String out\_trade\_no = WXRequestUtil.getTradeNo(prefix);
21. //实例化客户端
22. AlipayClient alipayClient = **new** DefaultAlipayClient("https://openapi.alipay.com/gateway.do
24. ",
25. AlipayUtil.APPID, AlipayUtil.PRIVATE\_KEY, "json", AlipayUtil.CHARSET, AlipayUtil.PUBLIC\_KEY, "RSA2");
26. //实例化具体API对应的request类,类名称和接口名称对应,当前调用接口名称：alipay.trade.app.pay
27. AlipayTradeAppPayRequest alirequest = **new** AlipayTradeAppPayRequest();
28. //SDK已经封装掉了公共参数，这里只需要传入业务参数。以下方法为sdk的model入参方式(model和biz\_content同时存在的情况下取biz\_content)。
29. AlipayTradeAppPayModel model = **new** AlipayTradeAppPayModel();
30. model.setPassbackParams(URLEncoder.encode(body.toString()));;  //描述信息  添加附加数据
31. model.setSubject(subject); //商品标题
32. model.setOutTradeNo(out\_trade\_no); //商家订单编号
33. model.setTimeoutExpress("30m"); //超时关闭该订单时间
34. model.setTotalAmount(totalAmount);  //订单总金额
35. model.setProductCode("QUICK\_MSECURITY\_PAY"); //销售产品码，商家和支付宝签约的产品码，为固定值QUICK\_MSECURITY\_PAY
36. alirequest.setBizModel(model);
37. alirequest.setNotifyUrl("http://pay.gzdxfszx.com/pay/alipay/notify.xs
39. ");  //回调地址
40. String orderStr = "";
41. **try** {
42. //这里和普通的接口调用不同，使用的是sdkExecute
43. AlipayTradeAppPayResponse response = alipayClient.sdkExecute(alirequest);
44. orderStr = response.getBody();
45. //                System.out.println(orderStr);//就是orderString 可以直接给客户端请求，无需再做处理。
46. } **catch** (AlipayApiException e) {
47. e.printStackTrace();
48. }
50. String sign = MD5Encode.encode(out\_trade\_no);
51. savePayLog(out\_trade\_no,totalAmount,sid,body, sign,nid,1);
53. JSONObject resObj = **new** JSONObject();
54. resObj.put("code",ResponseCode.HTTP\_SUCCESS);
55. resObj.put("out\_trade\_no",out\_trade\_no);
56. resObj.put("sign",sign);
57. resObj.put("orderStr",orderStr);
59. res = resObj.toJSONString();
60. }**catch** (Exception e){
61. e.printStackTrace();
62. }
63. **return** res;
64. }

异步通知处理

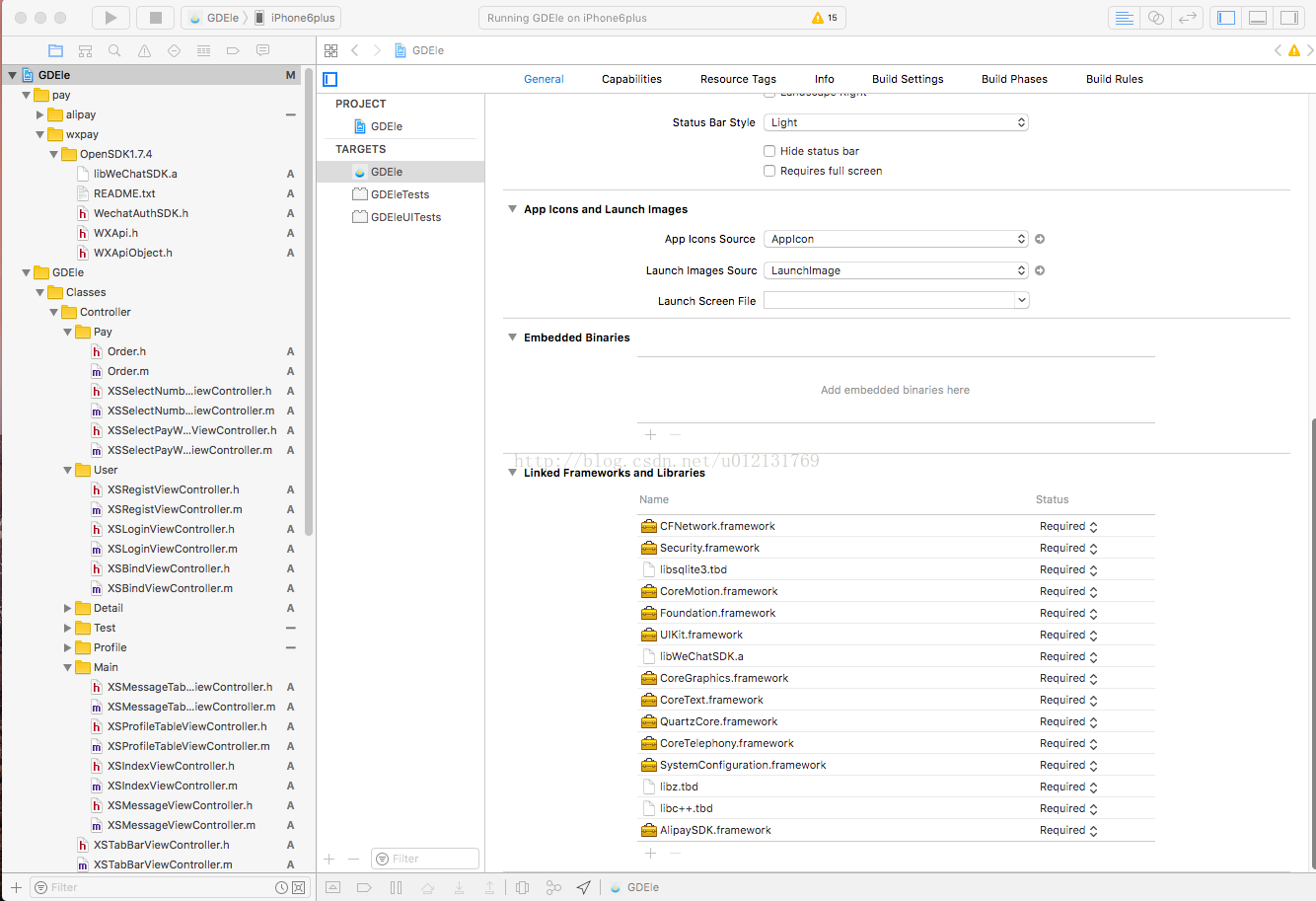
**[java]** [view plain](http://blog.csdn.net/u012131769/article/details/76639527) [copy](http://blog.csdn.net/u012131769/article/details/76639527)

[print?](http://blog.csdn.net/u012131769/article/details/76639527)

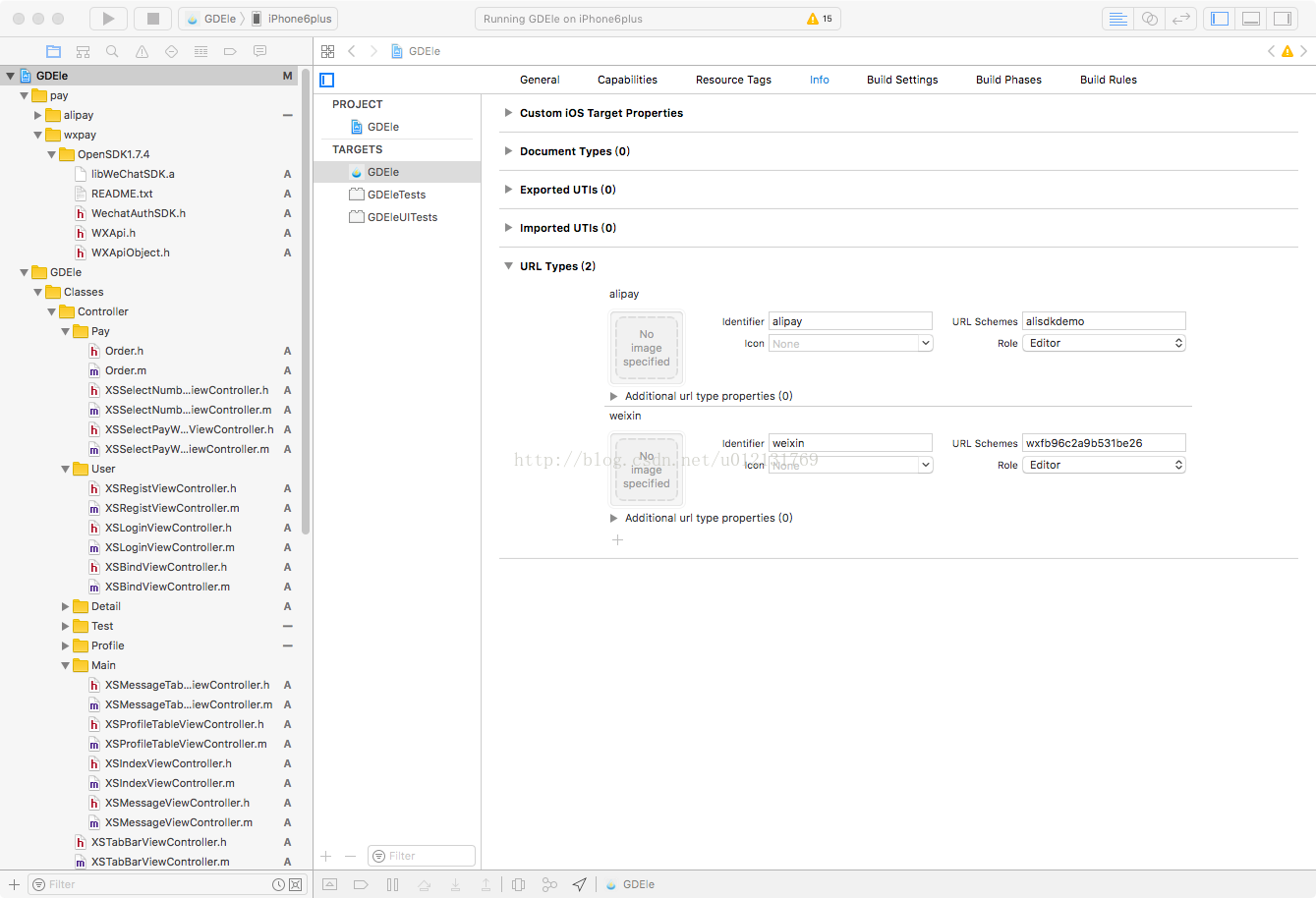
1. @RequestMapping(value="/pay/alipay/notify",produces="application/json;charset=utf-8")
2. @ResponseBody
3. **public** String notify(HttpServletRequest request){
4. Map requestParams = request.getParameterMap();
5. System.out.println("支付宝支付结果通知"+requestParams.toString());
6. //获取支付宝POST过来反馈信息
7. Map<String,String> params = **new** HashMap<String,String>();
9. **for** (Iterator iter = requestParams.keySet().iterator(); iter.hasNext();) {
10. String name = (String) iter.next();
11. String[] values = (String[]) requestParams.get(name);
12. String valueStr = "";
13. **for** (**int** i = 0; i < values.length; i++) {
14. valueStr = (i == values.length - 1) ? valueStr + values[i]
15. : valueStr + values[i] + ",";
16. }
17. //乱码解决，这段代码在出现乱码时使用。
18. //valueStr = new String(valueStr.getBytes("ISO-8859-1"), "utf-8");
19. params.put(name, valueStr);
20. }
21. //切记alipaypublickey是支付宝的公钥，请去open.alipay.com
23. 对应应用下查看。
24. //boolean AlipaySignature.rsaCheckV1(Map<String, String> params, String publicKey, String charset, String sign\_type)
25. **try** {
26. //验证签名
27. **boolean** flag = AlipaySignature.rsaCheckV1(params, AlipayUtil.PUBLIC\_KEY, AlipayUtil.CHARSET, "RSA2");
28. **if**(flag){
29. **if**("TRADE\_SUCCESS".equals(params.get("trade\_status"))){
30. //                    //付款金额
31. //                    String amount = params.get("buyer\_pay\_amount");
32. //                    //支付宝交易号
33. //                    String trade\_no = params.get("trade\_no");
34. //                    //附加数据
35. //                    String passback\_params = URLDecoder.decode(params.get("passback\_params"));
37. //商户订单号
38. String out\_trade\_no = params.get("out\_trade\_no");
39. updateOrderInfo(out\_trade\_no,MD5Encode.encode(out\_trade\_no));
40. }
41. }
42. } **catch** (AlipayApiException e) {
43. // TODO Auto-generated catch block
44. e.printStackTrace();
45. }
46. **return** "success";
47. }

## 3.****IOS客户端开发****

导入支付宝开发包



添加URL Types



在需要调用支付宝的地方实现方法

**[objc]** [view plain](http://blog.csdn.net/u012131769/article/details/76639527) [copy](http://blog.csdn.net/u012131769/article/details/76639527)

[print?](http://blog.csdn.net/u012131769/article/details/76639527)

1. -(**void**)aliPay
2. {
3. [MBProgressHUD showMessage:@"跳转中，请稍候"];
4. **NSString** \*appScheme = @"alisdkdemo";
6. **NSString** \*userUrlStr = [NSString stringWithFormat:@"%@?sid=%@&account=%@&desc=%@", ALI\_PREPAY\_URL, **self**.student.sid,**self**.payJinE,**self**.student.nid];
7. **NSURL** \*url = [NSURL URLWithString:userUrlStr];
8. //     NSLog(@"userUrlStr = %@", userUrlStr);
10. **NSURLRequest** \*request = [NSURLRequest requestWithURL:url];
11. **AFHTTPRequestOperation** \*operation = [[AFHTTPRequestOperation alloc]initWithRequest:request];
13. [operation setCompletionBlockWithSuccess:^(**AFHTTPRequestOperation** \*operation, **NSDictionary** \*responseObject) {
14. [MBProgressHUD hideHUD];
16. //                 NSLog(@"支付宝支付的response = %@", operation.responseString);
17. **NSData** \*JSONData = [operation.responseString dataUsingEncoding:NSUTF8StringEncoding];
18. **NSDictionary** \*userDict = [NSJSONSerialization JSONObjectWithData:JSONData options:NSJSONReadingMutableLeaves error**:nil**];
20. **NSString** \*orderString = [userDict objectForKey:@"orderStr"];
21. **self**.sign = [userDict objectForKey:@"sign"];
22. **self**.ordnum = [userDict objectForKey:@"out\_trade\_no"];
24. [[AlipaySDK defaultService] payOrder:orderString fromScheme:appScheme callback:^(**NSDictionary** \*resultDic) {
25. NSLog(@"reslut = %@",resultDic);
26. [MBProgressHUD hideHUD];
28. **NSString** \*resultStatus = [resultDic objectForKey:@"resultStatus"];
29. **if**([resultStatus isEqualToString:@"9000"]){
30. // 支付成功，更新用户信息
31. [**self** payDidFinish];
32. }**else**{
33. // 支付失败，
34. [**self** setupAlertControllerWithTitle:@"支付宝支付结果" messge:@"本次支付未完成，您可以稍后重试！" confirm:@"好的"];
35. }
37. }];
38. }failure:^(**AFHTTPRequestOperation** \*operation, **NSError** \*error) {
39. [MBProgressHUD hideHUD];
40. NSLog(@"发生错误！%@",error);
41. }];
42. **NSOperationQueue** \*queue = [[NSOperationQueue alloc] init];
43. [queue addOperation:operation];


47. }

银联支付: <https://open.unionpay.com/ajweb/help/file/techFile?productId=3>

Paypal支付: <https://developer.paypal.com/docs/>