剛谷科技股份有限公司

GanguTech Co., Ltd.

萬付通-代收代付平台

APP SDK 介接文件

Revision and Sign Off Sheet

Change Record:

Date	Version	Change Reason
2015-04-29	1.0.0	
2015-08-19	1.0.1	1.增加錯誤代碼清單 2.增加支付方式代碼
2015-08-27	1.0.2	1.增加錯誤代碼清單(20033,20034)
2015-12-03	1.0.3	1.更新信用卡付款呼叫方式 2.增加內建 UI 與呼叫方式提供使用 3.增加彰化銀行實體 ATM 付款提示建議

目錄

一、 系統需求

iOS SDK 7.0+

Android 4.0+

二、 主旨

OnePaid 代收代付功能 APP SDK 介接文件，提供特約商店依照本文件進行 APP 金流付款功能介接。

三、 主要提供服務

本文件主要提供特約商店五種服務，分別為如下所述：

1. 取得可用金流

提供特約商店取得可使用之金流列表。

2. 使用信用卡付款

提供特約商店讓消費者進行信用卡付款行為，卡號輸入將於 SDK 進行。

3. 使用 ATM 或超商付款

提供特約商店讓消費者進行 ATM 或超商付款行為。

若使用者選擇彰化銀行實體 ATM 進行付款，建議提示使用者「選擇彰化銀行實體 ATM 付款，請您至自動櫃員機操作時，選擇『繳費』選項，謝謝」。

4. 更新訂單狀態

提供特約商店呼叫本服務進行訂單確認，當呼叫成功後本系統將不再發送訂單付款通知

5. 由 SDK 進行相關付款操作

提供特約商店呼叫本服務，進行由選擇付款方式至付款結果流程

四、 安全性防護政策

1. 使用 SSL 憑證

本系統傳輸資料均使用 SSL 憑證進行資料傳輸加密，防止傳輸中資料被惡意竊取。

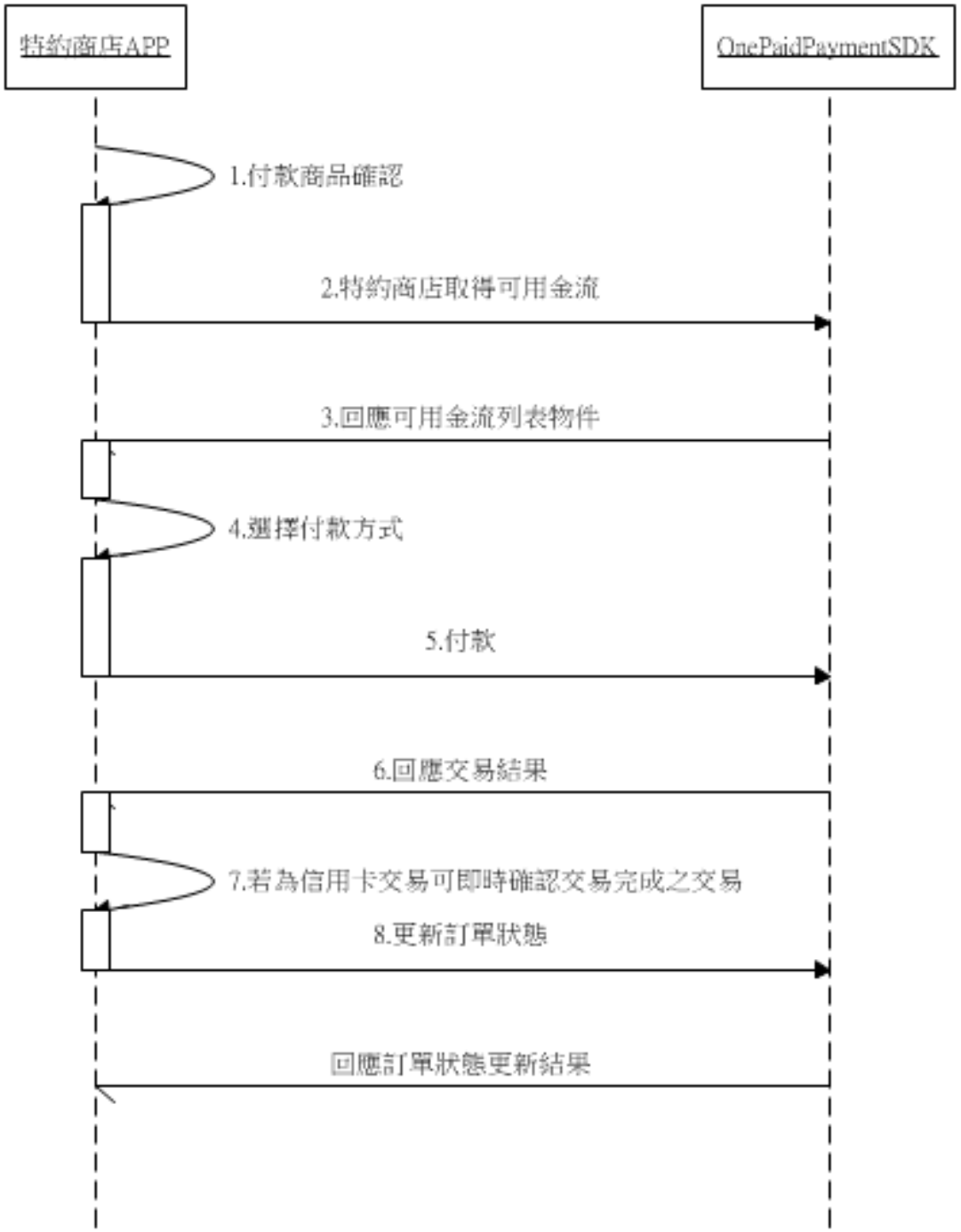
2. 資料傳輸使用檢核碼機制

本系統於資料傳輸發送及回應時需要將資料參數與商家金鑰進行串組產出一組 MD5 字串，本系統與商家系統應該以相同邏輯針對此檢核碼進行驗證，以確保交易正確性。

五、 主動通知服務之規則

主動通知服務為使用非即時交易，需通知已付款時之對應服務，為 Server to Server 之服務，詳情請參考『OnePaid 代收代付 API 介接文件』

六、 交易流程意示圖


特約商店APP
OnePaidPaymentSDK
1.付款商品確認
2.特約商店取得可用金流
3.回應可用金流列表物件
4.選擇付款方式
5.付款
6.回應交易結果
7.若為信用卡交易可即時確認交易完成之交易
8.更新訂單狀態
回應訂單狀態更新結果

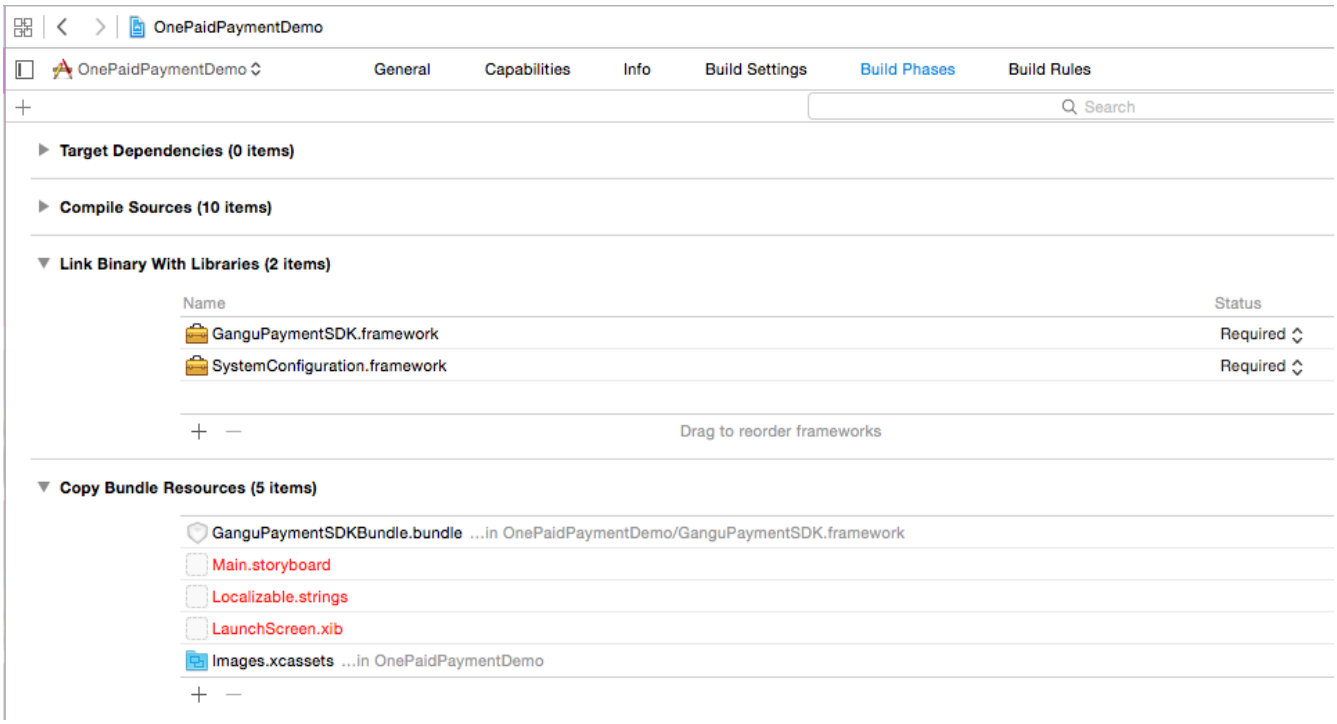
七、 SDK 服務說明

1. Framework 設定

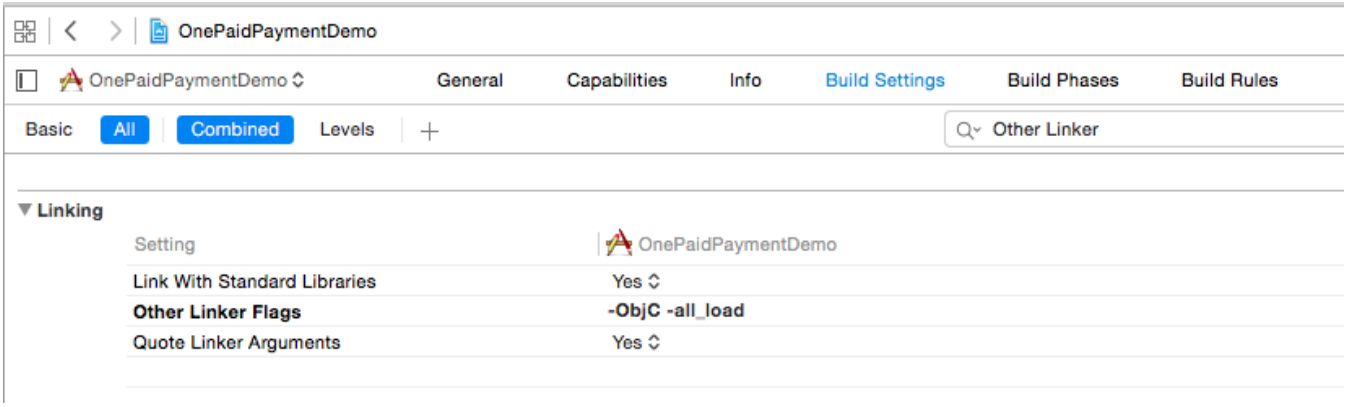
- **[iOS]**

Build Phases > Link Binary With Libraries 新增 GanguPaymentSDK.framework 與 SystemConfiguration.framework

Build Phases > Copy Bundle Resources 新增 GanguPaymentSDKBundle.bundle(此 bundle 檔案於 framework 內)

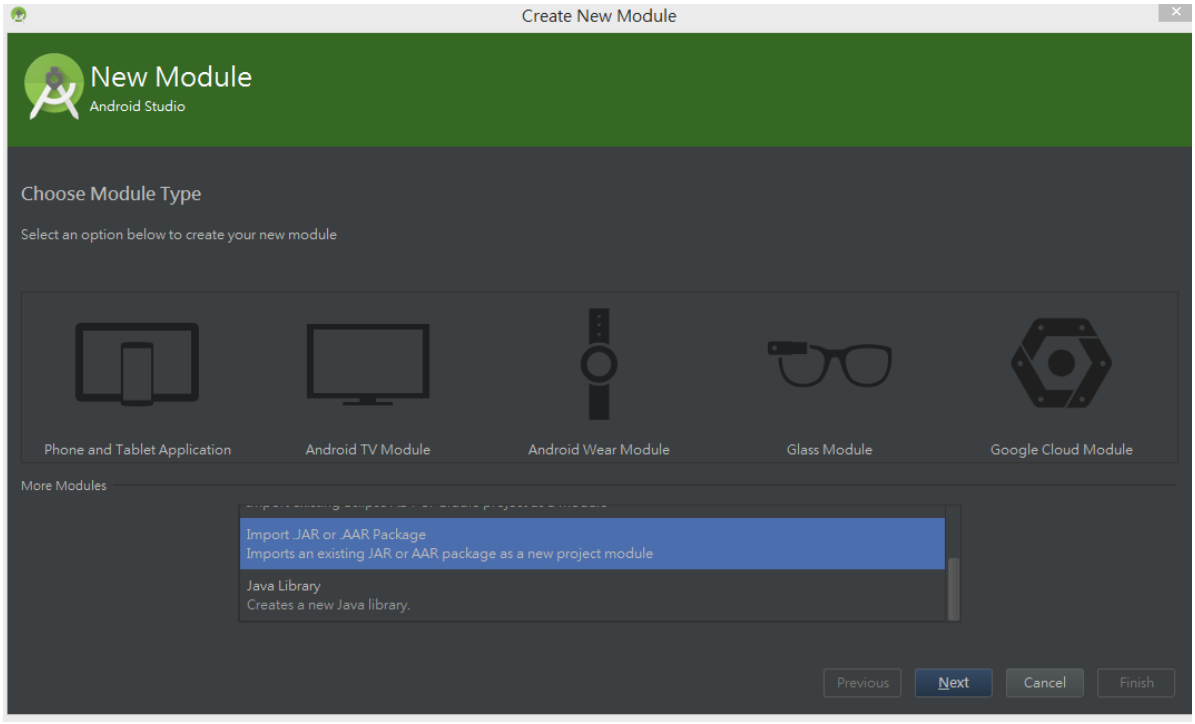


Build Settings > Other Linker Flags 新增 -ObjC 設定，若仍無法正常使用請再新增 –all_load 設定

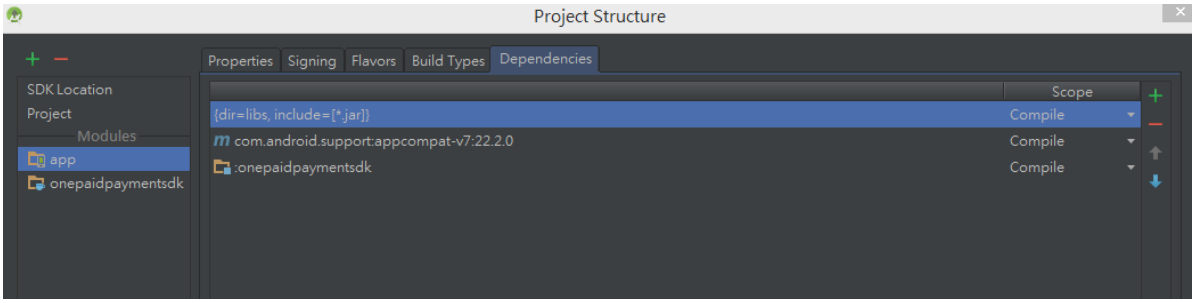


- **[Android]**

將 onepaidpaymentsdk.aar 檔放入 APP 欲放入 AAR 的資料夾中，選擇 File->New Module->import JAR or AAR Package。



將 onepaidpaymentsdk.aar 加入後，File->Project Structure->Dependencies 加入 onepaidpaymentsdk



2. SDK 設定

- **[iOS]**

使用 Delegate 做為資料回應方法，請務必設定並實作內容

```
#import <GanguPaymentSDK/GanguPaymentSDK.h>

@interface PreparePaymentViewController : UIViewController <GGTPaymentHandlerDelegate>
```

設定特約商店編號與安全碼及連線環境

```
// 建立PaymentSDK 物件
GGTPaymentHandler *paymentHandler = [GGTPaymentHandler sharedInstance];
// 設定商家ID與安全碼，連線至測試環境「若需連線至正是環境請設定official:YES」
[paymentHandler setConfigID:@"xxxxxxxxxxx" securityCode:@"xxxxxxxxxxxxxxxxxxxxx" official:NO];
// 設定PaymentSDK Delegate
[paymentHandler setDelegate:self];
```

- **[Android]**

初始化 OnePaidPayment 物件，並設定商店編號與安全碼

```
//設定 商家ID與安全碼 及連線環境
String MerKey = "XXXXXXXXXXXX";
String MerID = "XXXXXXXXXXX";
Boolean bCn=false;
//建立PaymentSDK物件
OnePaidPayment objOP= new OnePaidPayment(MainActivity.this, MerID, MerKey, bCn);
//設定PaymentSDK delegate
objOP.delegate = this;
```

3. 特約商店可用金流

當特約商店在進行交易之前，可使用此服務來跟本系統取得可用的金流服務

- **[iOS]**

呼叫取得可用金流：

```
// 呼叫取得可用金流列表
[[GGTPaymentHandler sharedInstance] getPaymentsList];
```

取得可用金流成功結果：

```
// 取得可用金流列表
- (void)getPaymentsListResult:(GGTPayments *)payments {
    NSLog(@"%@", payments);
}
```

- **[Android]**

呼叫取得可用金流：

```
objOP.getPaymentsList();
```

取得可用金流成功結果：

```
@Override
public void getPaymentsListResult(Payments p) {
    pm = p;
}
```

4. 使用信用卡付款

特約商店需將產生付款訂單的資訊傳送至本服務方法，並將畫面交給 SDK 進行後續付款動作。

【注意:訂單單號每次送出不可重複，不論成功或失敗】

- **[iOS]**

呼叫使用信用卡付款：

```
// 建立交易內容資訊
GGTTransactionInfo *transactionInfo = [[GGTTransactionInfo alloc] init];
[transactionInfo setMerTradeNo:@"Test123456789"];
[transactionInfo setMerTradeDate:[NSDate date]];
[transactionInfo setPaymentType:[NSNumber numberWithInt:5]];
[transactionInfo setProductName:@"Product Name"];
[transactionInfo setCurrencyType:[NSNumber numberWithInt:1]];
[transactionInfo setTotalAmt:@"100.00"];
[transactionInfo setRemark:@"Remark"];

UINavigationController *creditCardNavigation = [[GGTPaymentHandler sharedInstance] payUsingCreditCard:self.transactionInfo];
// 改變NavigationBar背景顏色
// [self.creditCardNavigation.navigationBar setBarTintColor:[UIColor redColor]];
// 改變NavigationBar文字顏色
// [self.creditCardNavigation.navigationBar setTintColor:[UIColor whiteColor]];

[self presentViewController:creditCardNavigation animated:YES completion:nil];
```

取得信用卡付款成功結果：

```
// 取得付款結果
- (void)payTransactionResult:(GGTTransactionResult *)result {
    NSLog(@"PaymentSDK Transaction Result : %@", result);
}
```

- **[Android]**

呼叫使用信用卡付款：

```
Bundle bData = this.getIntent().getExtras();
TransactionInfo ti = new TransactionInfo();
ti.CurrencyType = bData.getInt("CurrencyType");
ti.MerTradeDate = toolAPI.dStringtoDateTime(bData.getString("MerTradeDate"));
ti.MerTradeNo = bData.getString("MerTradeNo");
ti.PaymentType = bData.getInt("PaymentType");
ti.ProductName = bData.getString("ProductName");
ti.Remark = bData.getString("Remark");
ti.TotalAmt = bData.getString("TotalAmt");
objOP.delegate = this;
objOP.payUsingCreditCard(ti);
```

取得信用卡付款成功結果：

```
@Override
public void payUsingCreditCardResult(TransactionResult t) {
    tr=t;
}
```

5. 使用超商或實體 ATM 付款

特約商店需將產生付款訂單的資訊傳送至本服務方法，本服務將使用傳入資訊進行超商代碼、條碼或實體 ATM 轉帳帳號回傳，請務必將相關訊息顯示給使用者，提供使用者付款。

若使用者選擇彰化銀行實體 ATM 進行付款，建議提示使用者「選擇彰化銀行實體 ATM 付款，請您至自動櫃員機操作時，選擇『繳費』選項，謝謝」。

【注意:訂單單號每次送出不可重複，不論成功或失敗】

- **[iOS]**

呼叫使用超商或實體 ATM 付款

```
// 建立交易內容資訊
GGTTransactionInfo *transactionInfo = [[GGTTransactionInfo alloc] init];
[transactionInfo setMerTradeNo:@"Test123456789"];
[transactionInfo setMerTradeDate:[NSDate date]];
[transactionInfo setPaymentType:[NSNumber numberWithInt:4]];
[transactionInfo setProductName:@"Product Name"];
[transactionInfo setCurrencyType:[NSNumber numberWithInt:1]];
[transactionInfo setTotalAmt:@"100.00"];
[transactionInfo setRemark:@"Remark"];

[[GGTPaymentHandler sharedInstance] payUsingNonRealTimeWay:transactionInfo];
```

取得付款資訊結果

```
// 取得付款結果
- (void)payTransactionResult:(GGTTransactionResult *)result {
    NSLog(@"PaymentSDK Transaction Result : %@", result);
}
```

- **[Android]**

呼叫使用超商或實體 ATM 付款

```
Bundle bData = this.getIntent().getExtras();
TransactionInfo ti = new TransactionInfo();
ti.CurrencyType = bData.getInt("CurrencyType");
ti.MerTradeDate = toolAPI.dStringtoDateTime(bData.getString("MerTradeDate"));
ti.MerTradeNo = bData.getString("MerTradeNo");
ti.PaymentType = bData.getInt("PaymentType");
ti.ProductName = bData.getString("ProductName");
ti.Remark = bData.getString("Remark");
ti.TotalAmt = bData.getString("TotalAmt");
objOP.delegate = this;
objOP.payUsingNonRealTimeWay(ti);
```

取得付款資訊結果

```
@Override
public void payWithNonRealTimeWayResult(TransactionResult t) {

    tr = t;
}
```

6. SDK UI 進行付款

使用 SDK 內建 UI 提供由選擇付款方式，付款交易，付款結果呈現等頁面進行付款，**請注意，相關接收付款訊息方式仍應實作，若使用信用卡付款仍需呼叫「更新訂單狀態」方法**

- **[iOS]**

```
// 建立交易內容資訊
GGTTransactionInfo *transactionInfo = [[GGTTransactionInfo alloc] init];
[transactionInfo setMerTradeNo:@"Test123456789"];
[transactionInfo setMerTradeDate:[NSDate date]];
[transactionInfo setPaymentType:[NSNumber numberWithInt:4]];
[transactionInfo setProductName:@"Product Name"];
[transactionInfo setCurrencyType:[NSNumber numberWithInt:1]];
[transactionInfo setTotalAmt:@"100.00"];
[transactionInfo setRemark:@"Remark"];

UINavigationController *paymentSDKNavigation = [[GGTPaymentHandler sharedInstance] payWithUserInterface:self.transactionInfo];
// 改變NavigationBar背景顏色
// [self.creditCardNavigation.navigationBar setBarTintColor:[UIColor redColor]];
// 改變NavigationBar文字顏色
// [self.creditCardNavigation.navigationBar setTintColor:[UIColor whiteColor]];

[self presentViewController:paymentSDKNavigation animated:YES completion:nil];
```

```objc
// 取得交易異常訊息
- (void)paymentSDKFailWithError:(NSError *)error {
    NSLog(@"PaymentSDK Error : %@", error);
}

// 取得付款結果
- (void)payTransactionResult:(GGTTransactionResult *)result {
    NSLog(@"PaymentSDK Transaction Result : %@", result);
}
```

- **[Android]**

```
TransactionInfo ti = new TransactionInfo();
ti.MerTradeNo="1234567890";
ti.CurrencyType = 1;
ti.MerTradeDate =tradeDate;
ti.PaymentType = 0;
ti.ProductName = "TestItem";
ti.Remark = "TestItem Desc";
ti.TotalAmt = "100.00" ;
objOP.payWithUserInterface(ti);
```

7. 更新訂單狀態

當特約商店收到本系統的付款成功通知時，特約商店必須要呼叫此方法，當本系統收到特約商店呼叫時才將此筆訂單判定為商加已正確收到本系統的付款通知，而本系統將不會再針對此筆訂單發送後端通知給予特約商店，**請注意，若使用非在線即時付款之支付方式如實體 ATM、超商代碼等，請勿在呼叫『使用超商或實體 ATM 付款』時就呼叫此服務，應該在接收到後端 Server to Server 通知時再呼叫**。

- **[iOS]**

呼叫更新訂單狀態：

```objc
// 更新訂單狀態，MerTradeNo與付款訂單單號相同
GGTConfirmOrderInfo *confirmOrderInfo = [[GGTConfirmOrderInfo alloc] init];
[confirmOrderInfo setMerTradeNo:@"Test123456789"];
[[GGTPaymentHandler sharedInstance] confirmOrder:confirmOrderInfo];
```

取得更新訂單狀態成功結果

```objc
// 取得更新訂單狀態成功結果
- (void)confirmOrderResult:(GGTConfirmOrderResult *)result {
    NSLog(@"%@", result);
}
```

- **[Android]**

呼叫更新訂單狀態：

```
ConfirmOrderInfo coi = new ConfirmOrderInfo();
coi.MerTradeNo = "1234567890";
objOP.confirmOrder(coi);
```

取得更新訂單狀態成功結果

```
@Override
public void confirmOrderResult(ConfirmOrderResult c) {
    cr = c;
}
```

8. 交易異常訊息

- **[iOS]**

以上服務若有任何異常與交易失敗，皆以本方法通知

```
// 取得交易異常訊息
- (void)paymentSDKFailWithError:(NSError *)error {
    NSLog(@"PaymentSDK Error : %@", error);
}
```

- **[Android]**

請參閱回傳物件之 ResultCode 內容，對應訊息如訊息代碼清單。

八、 Data Object

1. Payments

備註：取得特店可用金流列表

參數名稱	參數說明	屬性	類型	備註
ATM	實體 ATM	Public	List<Payment>	
CreditCard	信用卡	Public	List<Payment>	
CVSCode	超商代碼	Public	List<Payment>	
CVSBarcode	超商條碼	Public	List<Payment>	

2. Payment

備註：金流項目內容

參數名稱	參數說明	屬性	類型	備註
PaywayType	付款類型	Public	Int	
PaymentType	付款方式	Public	Int	
PaymentName	付款名稱	Public	String	
PaymentImgUrl	圖片網址	Public	String	

3. TransactionInfo

備註：執行付款，使用者自行設定參數內容

參數名稱	參數說明	屬性	類型	驗證	備註
MerTradeNo	商家交易編號	Public	String	不為空，長度小於 20	
MerTradeDate	商家交易日期	Public	DateTime	不為空	
PaymentType	付款方式	Public	Int	不為 0	
ProductName	產品名稱	Public	String	不為空，長度小於 200	
CurrencyType	幣別	Public	Int	不為 0	
TotalAmt	總金額	Public	String	含小數點兩位之字串	小數點兩位 600.00
Remark	備註	Public	String	長度小於 300	

4. TransactionResult

參數名稱	參數說明	屬性	類型	驗證	備註
StatusCode	交易結果	Public	Int		
Message	交易訊息	Public	String		
MerID	商家編號	Public	String		
MerTradeNo	商家交易編號	Public	String		
TradeNo	OnePaid 交易編號	Public	String		
PaymentDate	付款時間	Public	DateTime		2015-03-08 15:42:39
PaywayType	付款類型	Public	Int		
PaymentType	付款方式	Public	Int		
ProductName	產品名稱	Public	String		
CurrencyType	幣別	Public	Int		請參考幣別代碼
TotalAmt	總金額	Public	String		小數點兩位 600.00
CustomsArgs	客製化參數	Public	String		
Remark	備註	Public	String		
SignCode	檢核碼	Public	String		
Data			Json(String)		平台錯誤內容

5. ConfirmOrderInfo

參數名稱	參數說明	屬性	類型	驗證	備註
MerTradeNo	商家交易編號	Public	String	不為空，長度小於 20	

6. ConfirmOrderResult

參數名稱	參數說明	屬性	類型	驗證	備註
StatusCode	交易結果	Public	Int		
Message	交易訊息	Public	String		

7. CustomsArgs

CustomArgsResult

備註：信用卡、超商、實體 ATM 交易回傳客製化參數

參數名稱	參數說明	屬性	類型	驗證	備註
VirtualAccount	虛擬帳號	Public	String		實體 ATM 使用
BankNo	銀行代號	Public	String		實體 ATM 使用
PayDeadline	繳費期限	Public	DateTime		實體 ATM、超商條碼、超商代碼使用
CreditCardNo	卡號	Public	String		信用卡使用
Barcode1	超商第一段條碼	Public	String		超商條碼使用
Barcode2	超商第二段條碼	Public	String		超商條碼使用
Barcode3	超商第三段條碼	Public	String		超商條碼使用
CVSCode	超商代碼	Pubilc	String		超商代碼使用
StoreName	超商名稱	Public	String		超商代碼使用

九、 支付類型代碼

代碼	描述
1	超商代碼
3	信用卡
4	實體 ATM
5	超商條碼

十、 支付方式代碼

以下範例僅供參考，各特約商店可用的支付方式請在交易之前，先呼叫【特約商店可用金流(CanUsePayment)】來取得

代碼(PaymentType)	描述(PaymentName)
4	全家超商代碼
5	信用卡
7	玉山實體 ATM
8	國泰實體 ATM
9	第一銀行實體 ATM
11	OK 超商代碼
13	土地銀行實體 ATM
16	華南實體 ATM
18	台企實體 ATM
20	台新實體 ATM
25	超商條碼
27	萊爾富超商代碼
29	兆豐實體 ATM
代碼(PaymentType)	描述(PaymentName) 測試環境 Only
22	不扣款實體 ATM 銀行
23	立即繳費超商
31	立即繳費條碼

十一、 幣別代碼

代碼	描述
1	新台幣(TWD)

十二、 訊息代碼清單

代碼	描述
10001	成功
10002	失敗
10003	建立訂單失敗
10004	訂單已存在
10005	轉跳付款畫面失敗
10006	處理中
10010	查無訂單資訊
20000	不正確的參數內容
20001	不正確的商家編號
20002	不正確的商家交易編號
20003	不正確的商家交易時間
20004	不正確的交易類型
20005	不正確的付款方式
20006	不正確的商品名稱
20007	不正確的幣別編號
20008	不正確的商品總金額
20009	不正確的商家回傳網址
20011	不正確的檢核碼
20012	不正確的訂單
20023	該筆交易已申請退款
20024	交易尚未完成，無法進行退款
20025	交易已超過 30 天

20026	此金流不允許退款
20027	不正確的退款理由
20028	確認退款時，訂單狀態有誤
20029	因已入帳金額餘額不足，故無法退款
20030	取得額度資料錯誤
20033	交易金額超過允許上限
20034	交易金額低於允許下限
300014	金流交易尚在處理中，請執行訂單查詢 API 重新查詢交易結果。(隔十分鐘)
300025	金流端回傳失敗(可能為輸錯信用卡號或餘額不足等由銀行端回傳的錯誤)
50001	服務無法使用
50002	服務無法存取
50003	服務暫停使用
59999	伺服器發生錯誤
70001	商家交易編號驗證錯誤
70002	交易日期驗證錯誤
70003	付款方是設定錯誤
70004	產品名稱驗證錯誤
70005	幣別驗證錯誤
70006	總金額驗證錯誤
70007	備註驗證錯誤
80001	信用卡卡號驗證錯誤
80002	安全碼驗證錯誤
80003	到期月驗證錯誤
80004	到期年驗證錯誤
90001	未設定商家 ID 與安全碼
90002	OnePaidPayment 連線失敗
90003	交易確認失敗