

歐付寶第三方支付股份有限公司

allPay Thrid-Party Payment Co., Ltd.

歐付寶金流整合 SDK 技術文件

allPay Payment Integration Specification

Version 1.0.12

2015/09/21

目錄

1. 修改歷程	1
2. 簡介	2
3. 介接測試相關資訊.....	2
4. 前置準備事項	3
4.1. 後台設定注意事項	3
4.2. 金流 SDK 元件.NET 版介接參考	4
4.3. 金流 SDK 元件 PHP 版介接參考	5
4.4. 金流 SDK 元件 JAVA 版介接參考	5
5. 類別清單	6
5.1. 全功能無履約保證類別(AllInOne Class).....	6
5.1.1. 訂單產生的方法(CheckOut/CheckOutString)	9
1. 一般產生訂單範例	14
2. ATM 付款方式有繳費期限範例	18
3. CVS 或 BARCODE 付款方式範例	23
4. Alipay 付款方式範例	28
5. Tenpay 付款方式範例	33
6. Credit 分期付款方式範例	37
7. Credit 定期定額付款方式範例	42
5.1.2. 取得付款結果通知的方法(CheckOutFeedback).....	47
5.1.2.1. 一般付款結果通知參數與範例	47
5.1.2.2. ATM 的取號結果通知參數與範例	54
5.1.2.3. CVS 或 BARCODE 的取號結果通知參數與範例	60
5.1.2.4. Credit 定期定額的授權成功通知參數與範例	67
5.1.3. 訂單查詢的方法(QueryTradeInfo)	74
5.1.3.1. 訂單查詢使用參數	75
5.1.3.2. 訂單查詢程式碼範例	77
5.1.4. 信用卡定期定額訂單查詢的方法(QueryPeriodCreditCardTradeInfo).....	89
5.1.4.1. 訂單查詢使用參數	89
5.1.4.2. 定期定額訂單查詢程式碼範例	90
5.1.5. 信用卡關帳/退刷/取消/放棄的方法(DoAction)	96

5.1.5.1.	信用卡訂單處理使用參數	97
5.1.5.2.	信用卡訂單處理程式碼範例	97
5.1.6.	廠商通知退款的方法(AioChargeback)	102
5.1.6.1.	廠商通知退款使用參數	102
5.1.6.2.	廠商通知退款程式碼範例	103
5.2.	全功能有履約保證類別(AllInEscrow Class)	107
5.2.1.	訂單產生的方法(CheckOut)	108
5.2.2.	取得付款結果通知的方法(CheckOutFeedback)	111
5.2.3.	通知出貨的方法(DeliveryNotify)	114
5.2.4.	接收商品異常/取消訂單通知的方法(OrderChangeFeedback)	116
5.2.5.	商品異常/取消訂單處理結果回覆的方法(OrderChangeNotify)	119
5.2.6.	接收訂單履約完成通知的方法(CompletedFeedback)	121
5.2.7.	訂單查詢的方法(CheckOut)	124
6.	參考資料	127
7.	附錄	127
7.1.	呼叫服務的方法(HttpMethod)	127
7.2.	付款方式(PaymentMethod)	127
7.3.	付款方式子項目(PaymentMethodItem)	128
7.4.	額外付款資訊(ExtraPaymentInfo)	129
7.5.	裝置類型(DeviceType)	129
7.6.	定期定額的週期種類(PeriodType)	129
7.7.	信用卡訂單處理動作(ActionType)	129
7.8.	訂單異動通知類型(NotifiedType)	129
7.9.	出貨狀態(ShippingState)	130
7.10.	電子發票開立註記(InvoiceState)	130
7.11.	電子發票載具類別(CarruerType)	130
7.12.	電子發票捐贈註記(Donation)	130
7.13.	電子發票列印註記(PrintMark)	130
7.14.	電子發票課稅類別(TaxType)	131
7.15.	電子發票通關方式(ClearanceMark)	131
7.16.	電子發票字軌類別(InvType)	131
7.17.	交易訊息代碼一覽表	131
7.18.	交易狀態代碼一覽表	132

1. 修改歷程

VERSION HISTORY

版本	修訂日期	修訂內容
1.0.0	2014/4/8	新建 New Release。
1.0.1	2014/4/29	<ul style="list-style-type: none">- 增加測試介接的 Key 與 IV。- 修改當付款方式為 ATM、CVS、BARCODE 時，PaymentInfoURL 為必填。
1.0.2	2014/5/20	<ul style="list-style-type: none">- 新增 PHP 範例程式碼。- 增加參考引用 SDK 說明。- 增加範例目錄。
1.0.3	2014/7/17	全功能無履約： <ul style="list-style-type: none">- 新增 IgnorePayment 參數，設定不顯示的付款方式。- 新增 ClientRedirectURL，當付款方式為 ATM、CVS 或 BARCODE 時，若有設定該參數，歐付寶會在瀏覽器端，傳送付款相關資訊於該參數設定的位址。
1.0.4	2014/7/24	全功能無履約： <ul style="list-style-type: none">- 修改 C#範例的錯誤顯示語法，由使用者自行判斷如何處理。
1.0.5	2014/9/30	<ul style="list-style-type: none">- 修改 PHP 範例程式碼。
1.0.6	2014/12/4	全功能無履約： <ul style="list-style-type: none">- 新增 Java 版本
1.0.7	2014/12/5	全功能無履約(不含 Java 版本)： <ul style="list-style-type: none">- 因應平台商合作機制，建立訂單增加 PlatformID 參數。- 新增信用卡定期定額訂單查詢 API。- 使用信用卡付款時，增加 Language 使用英文環境參數。- 增加平台商手續費參數。
1.0.8	2015/01/12	變更公司名稱。
1.0.9	2015/01/27	修改行動裝置的信用卡分期的處理方式。
1.0.10	2015/01/28	全功能無履約(Java 版本)： <ul style="list-style-type: none">- 因應平台商合作機制，建立訂單增加 PlatformID 參數。- 新增信用卡定期定額訂單查詢 API。- 使用信用卡付款時，增加 Language 使用英文環境參數。- 增加平台商手續費參數。
1.0.11	2015/09/07	全功能無履約(僅 PHP 版本)： <ul style="list-style-type: none">- 建立訂單增加電子發票相關參數。
1.0.12	2015/09/21	更新定期定額訂單查詢的方法 QueryCreditCardTradeInfo > QueryPeriodCreditCardTradeInfo

2. 簡介

歐付寶金流整合 SDK 技術文件(以下簡稱 SDK)，提供合作特店以及個人會員一個方便的介接方式，無須自行處理複雜的檢核，直接透過 SDK 提供的方法，便可以以較快速的方式介接歐付寶的金流系統。

3. 介接測試相關資訊

請使用以下資訊介接本公司金流服務，此為測試環境資訊，請勿將此資訊對正式環境做測試。測試無誤後，請將以下資訊更換成正式環境的相關資訊。

欄位	欄位說明
特店編號(MerchantID)	2000132
廠商後台測試環境	http://vendor-stage.allpay.com.tw ，此網站提供查詢測試的訂單相關資訊，也可執行訂單模擬付款的功能，送回付款通知的資訊到貴公司的網站。若執行模擬付款功能，無法收到付款通知時，請參考以下 <u>注意事項</u> 。
廠商後台測試帳號/密碼	StageTest/test1234
介接測試 Hash Key	5294y06JblSpM5x9
介接測試 Hash IV	v77hoKGq4kWxNNIS
信用卡測試卡號	4311-9522-2222-2222
信用卡測試安全碼	222
信用卡測試有效年月	請設定大於測試時間的年月。假如您的測試時間在 2013 年 11 月 26 號，該筆交易的信用卡有效年月請設定 2013 年 11 月以後，因為系統會判斷有效年月是否已過期，已過期則會回應刷卡失敗。

※注意事項:

註 1：若需要用到訂單查詢等幕後 POST 的功能，請於介接測試前，到「廠商後台 > 基本資料查詢 > 廠商基本資料」，設定貴公司伺服器的對外 IP 位址。(只有正式環境需要，測試環境不用設定)

註 2：傳值的參數中，名稱前有「*」表示必須帶此參數給 AllPay，若無，表示可不用帶此參數給 AllPay。

註 3：接收付款通知時，請系統開發人員確認下面事項，以利收到歐付寶發送的付款通知:

- (1) 請確認訂單產生的 ReturnURL 參數中的程式是否已開發完成。
- (2) 請確認 ReturnURL 中的網址，已經開放對外，從貴公司的網域之外也可以通知得到。
- (3) 請確認貴公司的 Server 是否有開通防火牆，以避免付款通知被防火牆阻擋。

4. 前置準備事項

4.1. 後台設定注意事項

請到「廠商後台 > 基本資料查詢 > 廠商基本資料」設定以下資料，以方便後續透過 SDK 進行金流介接。

1. 無論介接**有履約**或**無履約** SDK，請**務必填寫**「允許的 IP」。

因為有些方法為幕後作業，基於安全上的考量，AllPay 會屏蔽 IP 位址，所以請務必於該廠商後台功能**加入您伺服器的對外 IP 位址**。

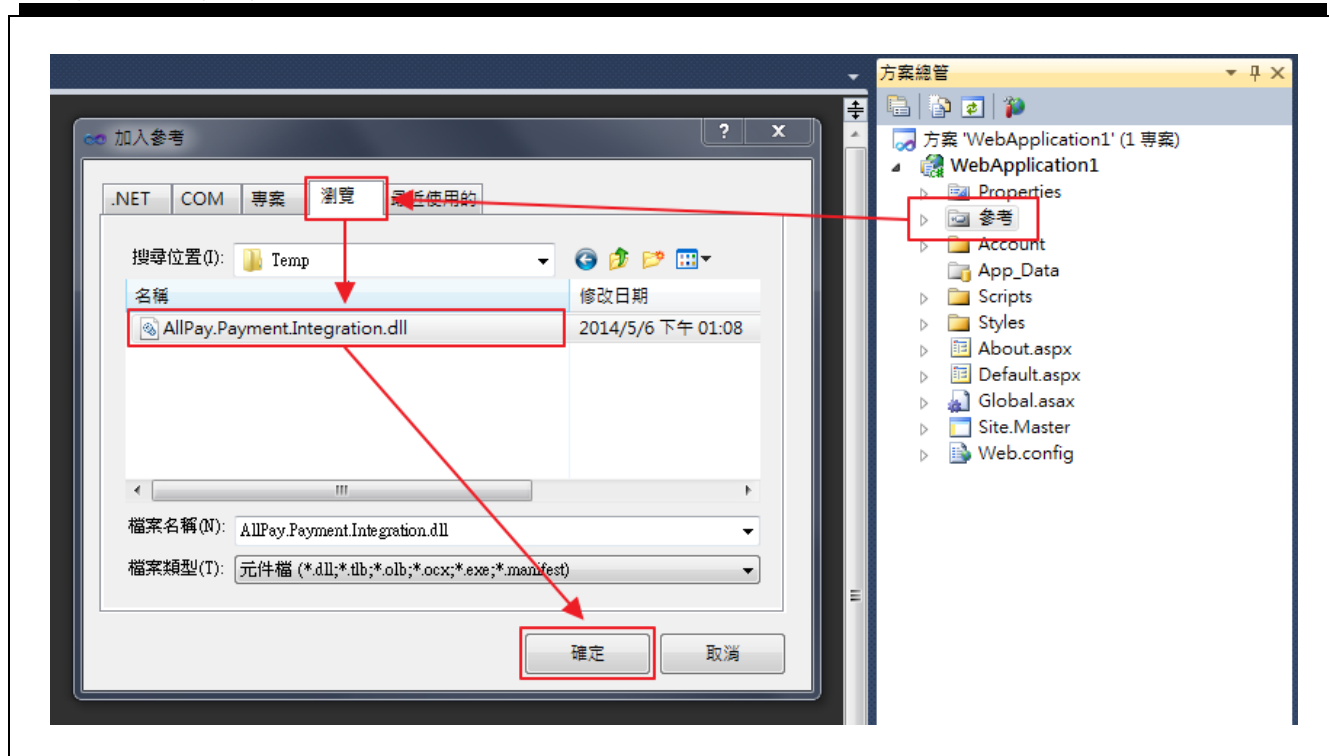
2. 當您介接**有履約**的 SDK，且想接收商品異常通知時，請務必填寫「接收商品異常網址」。
3. 當您介接**有履約**的 SDK，且想接收訂單取消通知時，請務必填寫「取消訂單回傳網址」。
4. 當您介接**有履約**的 SDK，且想接收履約完成通知時，請務必填寫「一般商品履約完成回傳網址」。

後台設定參考畫面如下圖：

基本資料	
公司登記名	歐付寶測試店家
統一編號	
負責人/個人姓名	歐○寶○試○家
連絡人姓名	
連絡人Email	
連絡人手機號碼	
網站名稱	
網站網址	http://www.allpay.com.tw
介接設定	
廠商編號	2000132
ALL IN ONE介接HashKey	
ALL IN ONE介接HashIV	
一般金流介接HashKey	
一般金流介接HashIV	
允許的IP	203.69.42.197##60.251.158.70##220.130.154.67##140.114.22 若有多組允許的IP，請用"##"做分隔，例：192.168.1.1##192.168.2.2
接收商品異常網址	http://dev.lecreuset.com.tw/allpay/connect/return/
取消訂單回傳網址	http://dev.lecreuset.com.tw/allpay/connect/return/1
一般商品履約完成回傳網址	http://dev.lecreuset.com.tw/allpay/connect/return/

4.2. 金流 SDK 元件.NET 版介接參考

當您下載該 SDK 元件後，要在.NET 專案中，使用 SDK 進行金流介接前，請於專案中參考該元件，方可在該專案中使用，如下圖：



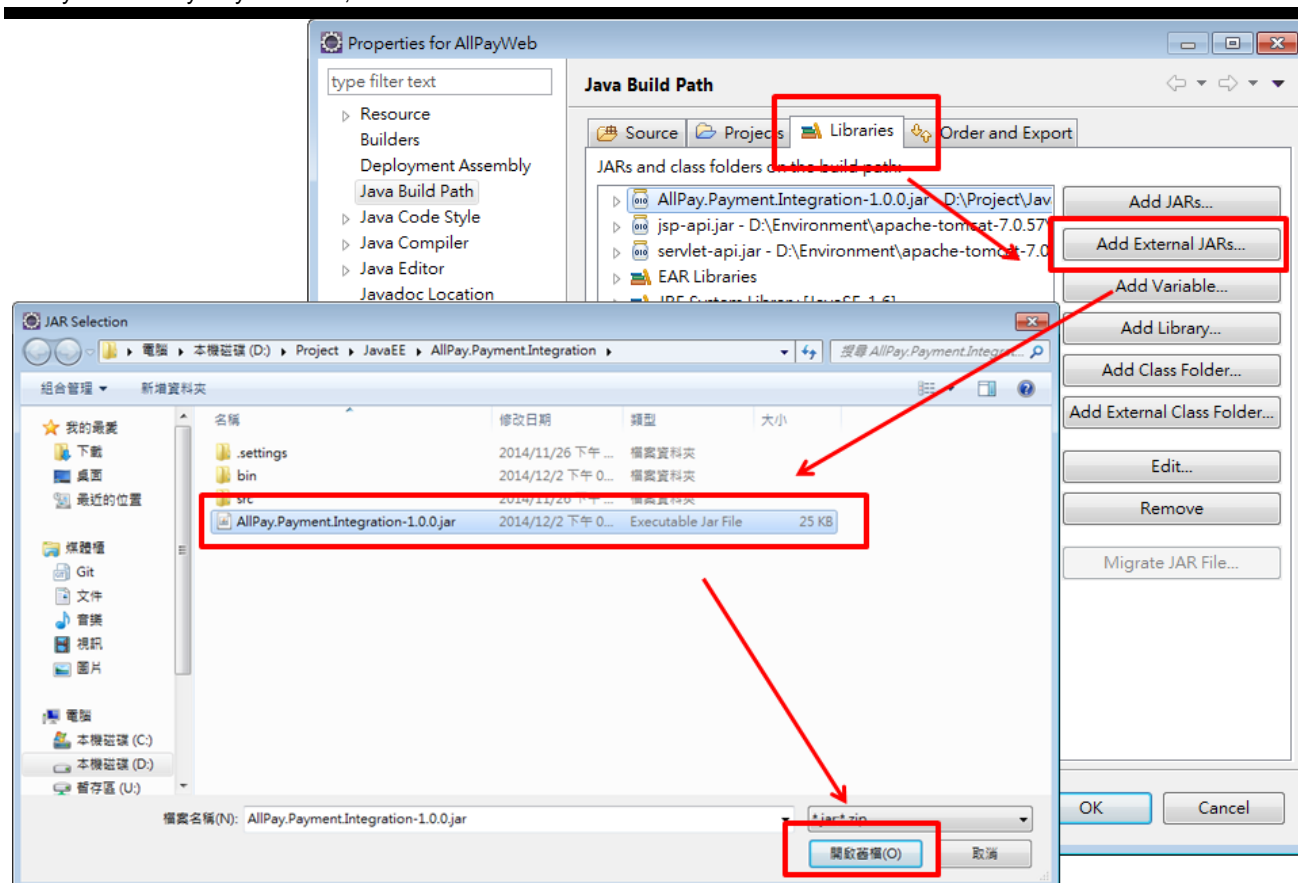
4.3. 金流 SDK 元件 PHP 版介接參考

當您下載該 SDK 元件後，要在 PHP 專案中，使用 SDK 進行金流介接前，請於專案中引用該元件，方可在該專案中使用，如下圖：



4.4. 金流 SDK 元件 JAVA 版介接參考


當您下載該 SDK 元件後，要在 JAVA 專案中，使用 SDK 進行金流介接前，請於專案中引用該元件，方可在該專案中使用，如下圖：



5. 類別清單

5.1. 全功能無履約保證類別(AllInOne Class)

該類別提供全功能無履約的介接方法，其資料傳送的參數成員如下表(名稱前若有紅色星號*為必填欄位)：

	名稱	型態	說明
	*ServiceURL	String(200)	要呼叫介接服務的網址。
	*ServiceMethod	HttpMethod	介接服務時，呼叫 API 的方法，預設為 HttpPOST。
	*HashKey	String	AllPay 提供的 HashKey。
	*HashIV	String	AllPay 提供的 HashIV。
	*MerchantID	String(10)	AllPay 提供的廠商編號。
	Send	SendArguments	介接產生訂單方法時，所需要的基本參數內容。
	SendExtend	SendExtendArguments	介接產生訂單方法時，所需要的延伸參數內容。

	Action	ActionArguments	介接信用卡關帳/退刷/取消/放棄時，所需要參數內容。
	ChargeBack	ChargeBackArguments	介接廠商退款時，所需要參數內容。
	Query	QueryArguments	介接訂單查詢時，所需要參數內容。
	CheckOut	IEnumerable<String>	產生訂單的方法(不支援 Ajax 非同步)。
	CheckOut(String)	IEnumerable<String>	於指定視窗產生訂單的方法(不支援 Ajax 非同步)。
	CheckOutString(ref String)	IEnumerable<String>	產生產生訂單 Html Code 的方法(支援 Ajax 非同步與 MVC)。
	CheckOutString(String, ref String)	IEnumerable<String>	於指定視窗產生產生訂單 Html Code 的方法(不支援 Ajax 非同步與 MVC)。
	CheckOutFeedback	IEnumerable<String>	取回訂單處理結果的方法。
	DoAction	IEnumerable<String>	信用卡關帳/退刷/取消/放棄的方法。
	AioChargeBack	IEnumerable<String>	廠商通知歐付寶退款的方法。
	QueryTradeInfo	IEnumerable<String>	查詢訂單資料的方法。
	QueryPeriodCreditCardTradeInfo	IEnumerable<String>	查詢定期定額訂單資料的方法。

建立該類別的 C#範例如下：

```

/*
 * 全功能介接的建構範例程式碼。
 */
try
{
    using (AllInOne oPayment = new AllInOne())
    {
        /* 服務參數 */
        oPayment.ServiceMethod = HttpMethod.HttpPOST;
        oPayment.ServiceURL = "<<您要呼叫的服務位址>>";
        oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";
        oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";
        oPayment.MerchantID = "<<AllPay提供給您的特店編號>>";
        .....
    }
}
catch (Exception ex)
{
    // 例外錯誤處理。
}
finally

```

```
{  
    // 顯示錯誤訊息。  
}
```

建立該類別的 PHP 範例如下：

```
/*  
 * 全功能介接的建構範例程式碼。  
 */  
$oPayment = new AllInOne();  
/* 服務參數 */  
$oPayment->ServiceMethod = HttpMethod::HttpPost;  
$oPayment->ServiceURL = "<<您要呼叫的服務位址>>";  
$oPayment->HashKey = "<<AllPay提供給您的Hash Key>>";  
$oPayment->HashIV = "<<AllPay提供給您的Hash IV>>";  
$oPayment->MerchantID = "<<AllPay提供給您的特店編號>>";  
.....
```

建立該類別的 JAVA 範例如下：

```
/*  
 * 全功能介接的建構範例程式碼。  
 */  
try {  
    AllInOne oPayment = new AllInOne();  
    /* 服務參數 */  
    oPayment.ServiceMethod = HttpMethod.HttpPost;  
    oPayment.ServiceURL = "<<您要呼叫的服務位址>>";  
    oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";  
    oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";  
    oPayment.MerchantID = "<<AllPay提供給您的特店編號>>";  
    //.....  
}  
catch (Exception e) {  
    // 例外錯誤處理。  
}  
finally {  
    // 顯示錯誤訊息。  
}
```

5.1.1. 訂單產生的方法(CheckOut/CheckOutString)





該方法僅適用於 Web 站台或 Web 專案，提供您使用 Http POST 的方式，於瀏覽器端送出訂單資料到 AllPay 的支付介面，進行訂單的付款。例：使用者在特店(賣家會員)消費後，特店(賣家會員)呼叫該方法，該方法會自動產生 POST 頁面，將訂單資料送至 AllPay 進行付款結帳作業，結帳完成後，AllPay 會將結果通知特店(賣家會員)，特店(賣家會員)可於接收頁面呼叫 [CheckOutFeedback 方法](#)，取回付款處理結果。

介接的服務網址(正式機為 https，測試機為 http)：

- 正式環境：<https://payment.allpay.com.tw/Cashier/AioCheckOut>
- 測試環境：<http://payment-stage.allpay.com.tw/Cashier/AioCheckOut>

基本資料傳送的參數成員如下表(名稱前若有紅色星號*為必填欄位)：






	名稱	型態	說明
	*Send.ReturnURL	String(200)	付款完成通知回傳的網址。
	Send.ClientBackURL	String(200)	瀏覽器端返回的廠商網址。 ※頁面導回的時候，不會帶付款結果到此網址，只是將頁面導回而已。
	Send.OrderResultURL	String(200)	瀏覽器端回傳付款結果網址。 ※設定了此參數值，會使設定的 ClientBackURL 失效。
	*Send.MerchantTradeNo	String(20)	廠商的交易編號，該交易編號不可重複。 ※該編號不可重複，所以不建議為訂單編號，建議產生該筆訂單要付款時使用的交易編號。
	*Send.MerchantTradeDate	DateTime	廠商的交易時間。
	*Send.TotalAmount	Decimal	交易總金額。
	*Send.TradeDesc	String(200)	交易描述。
	Send.ChoosePayment	PaymentMethod	使用的付款方式。(預設 ALL)
	Send.Remark	String(100)	備註欄位。
	Send.ChooseSubPayment	PaymentMethodItem	使用的付款子項目。(預設 None)
	Send.NeedExtraPaidInfo	ExtraPaymentInfo	是否需要額外的付款資訊。(預設 No)
	Send.DeviceSource	DeviceType	來源裝置。(預設 PC)
	Send.IgnorePayment	String(100)	不顯示的付款方式。 ※當 ChoosePayment 為 ALL 時，可選擇不顯示的付款方式，多筆請以井號分隔(#)；有效的參數

			值除 ALL 外，其餘請參考 PaymentMethod 。
	Send.PlatformID	String(10)	特約合作平台商代號(由 AllPay 提供)。 ※若 PlatformChargeFee 參數有帶值時，則 PlatformID 參數不可為空。
	Send.PlatformChargeFee	Integer	特約合作平台商手續費，一般廠商介接無須設定。
	*Send.Items	ItemCollection< Item >	訂單的商品資料。
	Send.InvoiceMark	InvoiceState	※僅 PHP 支援此功能 是否開立電子發票。(預設 No)






訂單商品資料傳送的成員如下表(名稱前若有紅色星號*為必填欄位)

	名稱	型態	說明
	*Name	String(20)	商品名稱。
	*Price	Decimal	商品單價。
	*Currency	String(10)	幣別單位。(例如：元)
	*Quantity	Integer	購買數量。
	URL	String(200)	商品的說明網址。

延伸資料傳送的參數成員如下表(名稱前若有紅色星號*為必填欄位)：

當電子發票開立註記設定為 Yes 時			
	名稱	型態	說明
	*SendExtend.RelateNumber	String(30)	※僅 PHP 支援此功能 廠商自訂編號，不可重複。
	SendExtend.CustomerIdentifier	String(8)	※僅 PHP 支援此功能 統一編號，當此參數不為空字串時，請設定固定長度 8 碼的數字。
	SendExtend.CarruerType	CarruerType	※僅 PHP 支援此功能 載具類別。 統一編號不為空字串時，載具類別請設定空字串。
	SendExtend.CustomerID	String(20)	※僅 PHP 支援此功能 客戶代號，僅可使用大小寫英文與數字。 當載具類別為會員載具(Member)時，此參數不可為空字串。
	SendExtend.Donation	Donation	※僅 PHP 支援此功能 捐贈註記，預設為不捐贈(No)。 統一編號不為空字串時，請設定不捐贈(No)。

	SendExtend.Print	PrintMark	<p>※僅 PHP 支援此功能</p> <p>列印註記，預設為不列印(No)。</p> <p>捐贈註記為捐贈(Yes)時，請設定不列印(No)。</p> <p>統一編號不為空字串時，請設定列印(Yes)。</p> <p>載具類別為會員載具(Member)、買受人自然人憑證(Citizen)、買受人手機條碼(Cellphone)時，請設定不列印(No)。</p>
	SendExtend.CustomerName	String(20)	<p>※僅 PHP 支援此功能</p> <p>客戶名稱，僅可使用中文、大小寫英文與數字。</p> <p>列印註記為列印(Yes)時，此參數不可為空字串。</p>
	SendExtend.CustomerAddr	String(200)	<p>※僅 PHP 支援此功能</p> <p>客戶地址。</p> <p>列印註記為列印(Yes)時，此參數不可為空字串。</p>
	SendExtend.CustomerPhone	String(20)	<p>※僅 PHP 支援此功能</p> <p>客戶手機號碼。</p> <p>此參數與客戶電子信箱擇一不可為空字串。</p>
	SendExtend.CustomerEmail	String(200)	<p>※僅 PHP 支援此功能</p> <p>客戶電子信箱。</p> <p>此參數與客戶手機號碼擇一不可為空字串。</p>
	*SendExtend.TaxType	TaxType	<p>※僅 PHP 支援此功能</p> <p>課稅類別。</p>
	SendExtend.ClearanceMark	ClearanceMark	<p>※僅 PHP 支援此功能</p> <p>通關方式。</p> <p>請設定空字串，僅課稅類別為零稅率(Zero)時，此參數不可為空字串。</p>
	SendExtend.CarruerNum	String(64)	<p>※僅 PHP 支援此功能</p> <p>載具編號。</p> <p>載具類別為無載具(None)或會員載具(Member)時，請設定空字串。</p> <p>載具類別為買受人自然人憑證(Citizen)時，請設定自然人憑證號碼，前 2 碼為大小寫英文，後 14 碼為數字。</p> <p>載具類別為買受人手機條碼(Cellphone)時，請設定手機條碼，第 1 碼為「/」，後 7 碼為大小寫英文、數字、「+」、「-」或「.」。</p>

	SendExtend.LoveCode	String(7)	※僅 PHP 支援此功能 愛心碼。 捐贈註記為捐贈(Yes)時，參數長度固定 3~7 碼，請設定全數字或第 1 碼大小寫「X」，後 2~6 碼全數字。
	*SendExtend.InvoiceItems	InvoiceItems Collection< InvoiceItems >	※僅 PHP 支援此功能 電子發票的商品資料。
	SendExtend.InvoiceRemark	String(Max)	※僅 PHP 支援此功能 備註。
	*SendExtend.DelayDay	Integer	※僅 PHP 支援此功能 延遲天數，範圍 0~15，設定為 0 時，付款完成後立即開立發票。
	*SendExtend.InvType	InvType	※僅 PHP 支援此功能 字軌類別。

電子發票商品資料傳送的成員如下表(名稱前若有紅色星號*為必填欄位)



	名稱	型態	說明
	*InvoiceItemName	String(Max)	商品名稱。
	*InvoiceItemCount	String(Max)	商品數量。
	*InvoiceItemWord	String(Max)	商品單位。
	*InvoiceItemPrice	String(Max)	商品價格。
	*InvoiceItemTaxType	String(Max)	商品課稅別(TaxType)。 課稅類別為混合(Mix)時，商品課稅別至少包含應稅(Dutiable)與免稅(Free)各 1 筆。

當付款方式設定為 ATM 時

	名稱	型態	說明
	SendExtend.ExpireDate	Integer	允許繳費的有效天數，最長 60 天，最短 1 天。(預設 3 天)
	*SendExtend.PaymentInfoURL	String(200)	伺服器端回傳付款相關資訊。
	SendExtend.ClientRedirectURL	String(200)	此網址為訂單建立完成後(非付款完成後)，歐付寶會將付款相關資訊以 Client 端方式回傳給特約商店，參數資訊與 PaymentInfoURL 相同。※有設定此參數 ClientBackURL 參數將會失去作用。

當付款方式設定為 CVS 或 BARCODE 時

	名稱	型態	說明
	SendExtend.Desc_1	String(20)	交易描述 1，會顯示在超商繳費平台的螢幕上。
	SendExtend.Desc_2	String(20)	交易描述 2，會顯示在超商繳費平台的螢幕上。
	SendExtend.Desc_3	String(20)	交易描述 3，會顯示在超商繳費平台的螢幕上。
	SendExtend.Desc_4	String(20)	交易描述 4，會顯示在超商繳費平台的螢幕上。
	*SendExtend.PaymentInfoURL	String(200)	伺服器端回傳付款相關資訊。
	SendExtend.ClientRedirectURL	String(200)	此網址為訂單建立完成後(非付款完成後)，歐付寶會將付款相關資訊以 Client 端方式回傳給特約商店，參數資訊與 PaymentInfoURL 相同。 ※有設定此參數 ClientBackURL 參數將會失去作用。
當付款方式設定為 Alipay 時			
	名稱	型態	說明
	*SendExtend.Email	String(200)	購買人信箱，支付寶記錄用，本公司不會記錄。
	*SendExtend.PhoneNo	String(20)	購買人電話，支付寶記錄用，本公司不會記錄。
	*SendExtend.UserName	String(20)	購買人姓名，支付寶記錄用，本公司不會記錄。
當付款方式設定為 Tenpay 時			
	名稱	型態	說明
	SendExtend.ExpireTime	DateTime	付款截止時間，只能帶入送出交易後的 72 小時。 (預設 72 小時)
當付款方式設定為 Credit，而且要變更畫面語系時			
	名稱	型態	說明
	SendExtend.Language	String(3)	語系設定。(預設 空[中文], ENG[英文])
當付款方式設定為 Credit，而且要使用分期付款時(不可與定期定額參數一同使用)			
	名稱	型態	說明
	SendExtend.CreditInstallment	Integer	刷卡分期期數，預設不提供分期。(預設 0)
	SendExtend.InstallmentAmount	Decimal	刷卡分期的付款金額，預設不提供分期。(預設 0)
	SendExtend.Redeem	Boolean	是否使用紅利折抵，使用時請設定 True。
	SendExtend.UnionPay	Boolean	是否是為聯銀卡，使用時請設定 True。
當付款方式設定為 Credit，而且要使用定期定額方式時(不可與分期付款參數一同使用)			
	名稱	型態	說明
	SendExtend.PeriodAmount	Decimal	每次授權金額。 ※當使用定期定期時，此參數必須要設定。
	SendExtend.PeriodType	PeriodType	週期種類。 ※當使用定期定期時，此參數必須要設定。
	SendExtend.Frequency	Integer	執行頻率。 ※當使用定期定期時，此參數必須要設定。

	SendExtend.ExecTimes	Integer	執行次數。 ※當使用定期定期時，此參數必須要設定。
	SendExtend.PeriodReturnURL	String(200)	伺服器端回傳定期定額的執行結果網址。

1. 一般產生訂單範例

一般產生訂單範例(C#)

```

/*
 * 產生訂單的範例程式碼。
 */
List<string> enErrors = new List<string>();

try
{
    using (AllInOne oPayment = new AllInOne())
    {
        /* 服務參數 */
        oPayment.ServiceMethod = HttpMethod.HttpPOST;
        oPayment.ServiceURL = "<<您要呼叫的服務位址>>";
        oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";
        oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";
        oPayment.MerchantID = "<<AllPay提供給您的特店編號>>";

        /* 基本參數 */
        oPayment.Send.ReturnURL = "<<您要收到付款完成通知的伺服器端網址>>";
        oPayment.Send.ClientBackURL = "<<您要歐付寶返回按鈕導向的瀏覽器端網址>>";
        oPayment.Send.OrderResultURL = "<<您要收到付款完成通知的瀏覽器端網址>>";
        oPayment.Send.MerchantTradeNo = "<<您此筆訂單交易編號>>";
        oPayment.Send.MerchantTradeDate = DateTime.Now("<<您此筆訂單的交易時間>>");
        oPayment.Send.TotalAmount = Decimal.Parse("<<您此筆訂單的交易總金額>>");
        oPayment.Send.TradeDesc = "<<您該筆訂單的描述>>";
        oPayment.Send.ChoosePayment = PaymentMethod.ALL;
        oPayment.Send.Remark = "<<您要填寫的其他備註>>";
        oPayment.Send.ChooseSubPayment = PaymentMethodItem.None;
        oPayment.Send.NeedExtraPaidInfo = ExtraPaymentInfo.No;
        oPayment.Send.DeviceSource = DeviceType.PC;
        oPayment.Send.IgnorePayment = "<<您不要顯示的付款方式>>"; // 例(排除支付寶與財富通): Alipay#Tenpay
        // 加入選購商品資料。
        oPayment.Send.Items.Add(new Item() {

```

```

        Name = "<<產品A>>", Price = Decimal.Parse("<<單價>>"), Currency = "<<幣別>>",
        Quantity = Int32.Parse("<<數量>>"), URL = "<<產品說明位址>>"
    });
    oPayment.Send.Items.Add(new Item() {
        Name = "<<產品B>>", Price = Decimal.Parse("<<單價>>"), Currency = "<<幣別>>",
        Quantity = Int32.Parse("<<數量>>"), URL = "<<產品說明位址>>"
    });
    oPayment.Send.Items.Add(new Item() {
        Name = "<<產品C>>", Price = Decimal.Parse("<<單價>>"), Currency = "<<幣別>>",
        Quantity = Int32.Parse("<<數量>>"), URL = "<<產品說明位址>>"
    });
    /* 產生訂單 */
    enErrors.AddRange(oPayment.CheckOut());
    /* 產生產生訂單 Html Code 的方法 */
    string szHtml = String.Empty;
    enErrors.AddRange(oPayment.CheckOutString(ref szHtml));
}
}
catch (Exception ex)
{
    // 例外錯誤處理。
    enErrors.Add(ex.Message);
}
finally
{
    // 顯示錯誤訊息。
    if (enErrors.Count() > 0)
        string szErrorMessage = String.Join("\r\n", enErrors);
}
}

```

一般產生訂單範例(PHP)

```

/*
 * 產生訂單的範例程式碼。
 */
try
{
    $oPayment = new AllInOne();
    /* 服務參數 */

```

```

$Payment->ServiceURL = "<<您要呼叫的服務位址>>";
$Payment->HashKey = "<<AllPay提供給您的Hash Key>>";
$Payment->HashIV = "<<AllPay提供給您的Hash IV>>";
$Payment->MerchantID = "<<AllPay提供給您的特店編號>>";
/* 基本參數 */
$Payment->Send['ReturnURL'] = "<<您要收到付款完成通知的伺服器端網址>>";
$Payment->Send['ClientBackURL'] = "<<您要歐付寶返回按鈕導向的瀏覽器端網址>>";
$Payment->Send['OrderResultURL'] = "<<您要收到付款完成通知的瀏覽器端網址>>";
$Payment->Send['MerchantTradeNo'] = "<<您此筆訂單交易編號>>";
$Payment->Send['MerchantTradeDate'] = date('Y/m/d H:i:s');
$Payment->Send['TotalAmount'] = (int) "<<您此筆訂單的交易總金額>>";
$Payment->Send['TradeDesc'] = "<<您該筆訂單的描述>>";
$Payment->Send['ChoosePayment'] = PaymentMethod::ALL;
$Payment->Send['Remark'] = "<<您要填寫的其他備註>>";
$Payment->Send['ChooseSubPayment'] = PaymentMethodItem::None;
$Payment->Send['NeedExtraPaidInfo'] = ExtraPaymentInfo::No;
$Payment->Send['DeviceSource'] = DeviceType::PC;
$Payment->Send['IgnorePayment'] = "<<您不要顯示的付款方式>>"; // 例(排除支付寶與財富通): Alipay#Tenpay
// 加入選購商品資料。
array_push($Payment->Send['Items'], array('Name' => "<<產品A>>", 'Price' => (int) "<<單價>>",
    'Currency' => "<<幣別>>", 'Quantity' => (int) "<<數量>>", 'URL' => "<<產品說明位址>>"));
array_push($Payment->Send['Items'], array('Name' => "<<產品B>>", 'Price' => (int) "<<單價>>",
    'Currency' => "<<幣別>>", 'Quantity' => (int) "<<數量>>", 'URL' => "<<產品說明位址>>"));
array_push($Payment->Send['Items'], array('Name' => "<<產品C>>", 'Price' => (int) "<<單價>>",
    'Currency' => "<<幣別>>", 'Quantity' => (int) "<<數量>>", 'URL' => "<<產品說明位址>>"));
/* 產生訂單 */
$Payment->Checkout();
/* 產生產生訂單 Html Code 的方法 */
$SzHtml = $Payment->CheckoutString();
}
catch (Exception $e)
{
    // 例外錯誤處理。
    throw $e;
}

```

一般產生訂單範例(JAVA)

```
/*
```

```
* 產生訂單的範例程式碼。
*/
List<String> enErrors = new ArrayList<String>();
try {
    AllInOne oPayment = new AllInOne();

    /* 服務參數 */
    oPayment.ServiceMethod = HttpMethod.HttpPOST;
    oPayment.ServiceURL = "<<您要呼叫的服務位址>>";
    oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";
    oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";
    oPayment.MerchantID = "<<AllPay提供給您的特店編號>>";

    /* 基本參數 */
    oPayment.Send.ReturnURL = "<<您要收到付款完成通知的伺服器端網址>>";
    oPayment.Send.ClientBackURL = "<<您要歐付寶返回按鈕導向的瀏覽器端網址>>";
    oPayment.Send.OrderResultURL = "<<您要收到付款完成通知的瀏覽器端網址>>";
    oPayment.Send.MerchantTradeNo = "<<您此筆訂單交易編號>>";
    oPayment.Send.MerchantTradeDate = new Date();/"<<您此筆訂單的交易時間>>"
    oPayment.Send.TotalAmount = new Decimal("<<您此筆訂單的交易總金額>>");
    oPayment.Send.TradeDesc = "<<您該筆訂單的描述>>";
    oPayment.Send.ChoosePayment = PaymentMethod.ALL;
    oPayment.Send.Remark = "<<您要填寫的其他備註>>";
    oPayment.Send.ChooseSubPayment = PaymentMethodItem.None;
    oPayment.Send.NeedExtraPaidInfo = ExtraPaymentInfo.No;
    oPayment.Send.DeviceSource = DeviceType.PC;
    oPayment.Send.IgnorePayment = "<<您不要顯示的付款方式>>"; // 例(排除支付寶與財富通): Alipay#Tenpay;

    // 加入選購商品資料。
    Item a1 = new Item();
    a1.Name = "<<產品A>>";
    a1.Price = new Decimal("<<單價>>");
    a1.Currency = "<<幣別>>";
    a1.Quantity = 0;/"<<數量>>"
    a1.URL = "<<產品說明位址>>";
    oPayment.Send.Items.add(a1);

    Item a2 = new Item();
```

```

        a2.Name = "<<產品B>>";
        a2.Price = new Decimal("<<單價>>");
        a2.Currency = "<<幣別>>";
        a2.Quantity = 0;//<<數量>>
        a2.URL = "<<產品說明位址>>";
        oPayment.Send.Items.add(a2);
        /* 產生訂單 */
        enErrors.addAll(oPayment.CheckOut(response.getWriter()));
        /* 產生產生訂單 Html Code 的方法 */
        StringBuilder szHtml = new StringBuilder();
        enErrors.addAll(oPayment.CheckOutString(szHtml));
    }
    catch (Exception e) {
        // 例外錯誤處理。
        enErrors.add(e.getMessage());
    }
    finally {
        // 顯示錯誤訊息。
        if (enErrors.size() > 0)
            out.print(enErrors);
    }
}

```

2. ATM 付款方式有繳費期限範例

ATM 付款方式有繳費期限範例(C#)

```

/*
 * 產生訂單的範例程式碼。
 */
List<string> enErrors = new List<string>();

try
{
    using (AllInOne oPayment = new AllInOne())
    {
        /* 服務參數 */
        oPayment.ServiceMethod = HttpMethod.HttpPOST;
        oPayment.ServiceURL = "<<您要呼叫的服務位址>>";
        oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";
    }
}

```

```
oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";
oPayment.MerchantID = "<<AllPay提供給您的特店編號>>";
/* 基本參數 */
oPayment.Send.ReturnURL = "<<您要收到付款完成通知的伺服器端網址>>";
oPayment.Send.ClientBackURL = "<<您要歐付寶返回按鈕導向的瀏覽器端網址>>";
oPayment.Send.OrderResultURL = "<<您要收到付款完成通知的瀏覽器端網址>>";
oPayment.Send.MerchantTradeNo = "<<您此筆訂單交易編號>>";
oPayment.Send.MerchantTradeDate = DateTime.Now("<<您此筆訂單的交易時間>>");
oPayment.Send.TotalAmount = Decimal.Parse("<<您此筆訂單的交易總金額>>");
oPayment.Send.TradeDesc = "<<您該筆訂單的描述>>";
oPayment.Send.ChoosePayment = PaymentMethod.ATM;
oPayment.Send.Remark = "<<您要填寫的其他備註>>";
oPayment.Send.ChooseSubPayment = PaymentMethodItem.None;
oPayment.Send.NeedExtraPaidInfo = ExtraPaymentInfo.No;
oPayment.Send.DeviceSource = DeviceType.PC;
// 加入選購商品資料。
oPayment.Send.Items.Add(new Item() {
    Name = "<<產品A>>", Price = Decimal.Parse("<<單價>>"), Currency = "<<幣別>>",
    Quantity = Int32.Parse("<<數量>>"), URL = "<<產品說明位址>>"
});
oPayment.Send.Items.Add(new Item() {
    Name = "<<產品B>>", Price = Decimal.Parse("<<單價>>"), Currency = "<<幣別>>",
    Quantity = Int32.Parse("<<數量>>"), URL = "<<產品說明位址>>"
});
oPayment.Send.Items.Add(new Item() {
    Name = "<<產品C>>", Price = Decimal.Parse("<<單價>>"), Currency = "<<幣別>>",
    Quantity = Int32.Parse("<<數量>>"), URL = "<<產品說明位址>>"
});
/* ATM 延伸參數 */
oPayment.SendExtend.ExpireDate = Int32.Parse("<<您允許的繳費有效天數>>");
oPayment.SendExtend.PaymentInfoURL = "<<您要收到付款相關資訊的伺服器端網址>>";
oPayment.SendExtend.ClientRedirectURL = "<<您要收到虛擬帳號相關資訊的伺服器端網址>>";
/* 產生訂單 */
enErrors.AddRange(oPayment.CheckOut());
/* 產生產生訂單 Html Code 的方法 */
string szHtml = String.Empty;
enErrors.AddRange(oPayment.CheckOutString(ref szHtml));
}
```

```

}
catch (Exception ex)
{
    // 例外錯誤處理。
    enErrors.Add(ex.Message);
}
finally
{
    // 顯示錯誤訊息。
    if (enErrors.Count() > 0)
        string szErrorMessage = String.Join("\r\n", enErrors);
}

```

ATM 付款方式有繳費期限範例(PHP)

```

/*
 * 產生訂單的範例程式碼。
 */
try
{
    $oPayment = new AllInOne();
    /* 服務參數 */
    $oPayment->ServiceURL = "<<您要呼叫的服務位址>>";
    $oPayment->HashKey = "<<AllPay提供給您的Hash Key>>";
    $oPayment->HashIV = "<<AllPay提供給您的Hash IV>>";
    $oPayment->MerchantID = "<<AllPay提供給您的特店編號>>";
    /* 基本參數 */
    $oPayment->Send['ReturnURL'] = "<<您要收到付款完成通知的伺服器端網址>>";
    $oPayment->Send['ClientBackURL'] = "<<您要歐付寶返回按鈕導向的瀏覽器端網址>>";
    $oPayment->Send['OrderResultURL'] = "<<您要收到付款完成通知的瀏覽器端網址>>";
    $oPayment->Send['MerchantTradeNo'] = "<<您此筆訂單交易編號>>";
    $oPayment->Send['MerchantTradeDate'] = date('Y/m/d H:i:s');
    $oPayment->Send['TotalAmount'] = (int) "<<您此筆訂單的交易總金額>>";
    $oPayment->Send['TradeDesc'] = "<<您該筆訂單的描述>>";
    $oPayment->Send['ChoosePayment'] = PaymentMethod::ATM;
    $oPayment->Send['Remark'] = "<<您要填寫的其他備註>>";
    $oPayment->Send['ChooseSubPayment'] = PaymentMethodItem::None;
    $oPayment->Send['NeedExtraPaidInfo'] = ExtraPaymentInfo::No;
    $oPayment->Send['DeviceSource'] = DeviceType::PC;
}

```

```
// 加入選購商品資料。
array_push($oPayment->Send['Items'], array('Name' => "<<產品A>>", 'Price' => (int)"<<單價>>",
    'Currency' => "<<幣別>>", 'Quantity' => (int) "<<數量>>", 'URL' => "<<產品說明位址>>"));
array_push($oPayment->Send['Items'], array('Name' => "<<產品B>>", 'Price' => (int)"<<單價>>",
    'Currency' => "<<幣別>>", 'Quantity' => (int) "<<數量>>", 'URL' => "<<產品說明位址>>"));
array_push($oPayment->Send['Items'], array('Name' => "<<產品C>>", 'Price' => (int)"<<單價>>",
    'Currency' => "<<幣別>>", 'Quantity' => (int) "<<數量>>", 'URL' => "<<產品說明位址>>"));
/* ATM 延伸參數 */
$oPayment->SendExtend['ExpireDate'] = (int) "<<您允許的繳費有效天數>>";
$oPayment->SendExtend['PaymentInfoURL'] = "<<您要收到付款相關資訊的伺服器端網址>>";
/* 產生訂單 */
$oPayment->CheckOut();
/* 產生產生訂單 Html Code 的方法 */
$sHtml = $oPayment->CheckOutString();
}
catch (Exception $e)
{
    // 例外錯誤處理。
    throw $e;
}
```

ATM 付款方式有繳費期限範例(JAVA)

```
/*
 * 產生訂單的範例程式碼。
 */
List<String> enErrors = new ArrayList<String>();
try {
    AllInOne oPayment = new AllInOne();

    /* 服務參數 */
    oPayment.ServiceMethod = HttpMethod.HttpPOST;
    oPayment.ServiceURL = "<<您要呼叫的服務位址>>";
    oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";
    oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";
    oPayment.MerchantID = "<<AllPay提供給您的特店編號>>";

    /* 基本參數 */
    oPayment.Send.ReturnURL = "<<您要收到付款完成通知的伺服器端網址>>";
```



```
oPayment.Send.ClientBackURL = "<<您要歐付寶返回按鈕導向的瀏覽器端網址>>";
oPayment.Send.OrderResultURL = "<<您要收到付款完成通知的瀏覽器端網址>>";
oPayment.Send.MerchantTradeNo = "<<您此筆訂單交易編號>>";
oPayment.Send.MerchantTradeDate = new Date();//"<<您此筆訂單的交易時間>>"
oPayment.Send.TotalAmount = new Decimal("<<您此筆訂單的交易總金額>>");
oPayment.Send.TradeDesc = "<<您該筆訂單的描述>>";
oPayment.Send.ChoosePayment = PaymentMethod.ATM;
oPayment.Send.Remark = "<<您要填寫的其他備註>>";
oPayment.Send.ChooseSubPayment = PaymentMethodItem.None;
oPayment.Send.NeedExtraPaidInfo = ExtraPaymentInfo.No;
oPayment.Send.DeviceSource = DeviceType.PC;

// 加入選購商品資料。
Item a1 = new Item();
a1.Name = "<<產品A>>";
a1.Price = new Decimal("<<單價>>");
a1.Currency = "<<幣別>>";
a1.Quantity = 0;//<<數量>>
a1.URL = "<<產品說明位址>>";
oPayment.Send.Items.add(a1);

Item a2 = new Item();
a2.Name = "<<產品B>>";
a2.Price = new Decimal("<<單價>>");
a2.Currency = "<<幣別>>";
a2.Quantity = 0;//<<數量>>
a2.URL = "<<產品說明位址>>";
oPayment.Send.Items.add(a2);

/* ATM 延伸參數 */
oPayment.SendExtend.ExpireDate = 0;//<<您允許的繳費有效天數>>
oPayment.SendExtend.PaymentInfoURL = "<<您要收到付款相關資訊的伺服器端網址>>";
oPayment.SendExtend.ClientRedirectURL = "<<您要收到虛擬帳號相關資訊的伺服器端網址>>";

/* 產生訂單 */
enErrors.addAll(oPayment.CheckOut(response.getWriter()));

/* 產生產生訂單 Html Code 的方法 */
StringBuilder szHtml = new StringBuilder();
enErrors.addAll(oPayment.CheckOutString(szHtml));
```

```
}  
catch (Exception e) {  
    // 例外錯誤處理。  
    enErrors.add(e.getMessage());  
}  
finally {  
    // 顯示錯誤訊息。  
    if (enErrors.size() > 0)  
        out.print(enErrors);  
}
```

3. CVS 或 BARCODE 付款方式範例

CVS 或 BARCODE 付款方式範例(C#)

```
/*  
 * 產生訂單的範例程式碼。  
 */  
List<string> enErrors = new List<string>();  
  
try  
{  
    using (AllInOne oPayment = new AllInOne())  
    {  
        /* 服務參數 */  
        oPayment.ServiceMethod = HttpMethod.HttpPOST;  
        oPayment.ServiceURL = "<<您要呼叫的服務位址>>";  
        oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";  
        oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";  
        oPayment.MerchantID = "<<AllPay提供給您的特店編號>>";  
        /* 基本參數 */  
        oPayment.Send.ReturnURL = "<<您要收到付款完成通知的伺服器端網址>>";  
        oPayment.Send.ClientBackURL = "<<您要歐付寶返回按鈕導向的瀏覽器端網址>>";  
        oPayment.Send.OrderResultURL = "<<您要收到付款完成通知的瀏覽器端網址>>";  
        oPayment.Send.MerchantTradeNo = "<<您此筆訂單交易編號>>";  
        oPayment.Send.MerchantTradeDate = DateTime.Now("<<您此筆訂單的交易時間>>");  
        oPayment.Send.TotalAmount = Decimal.Parse("<<您此筆訂單的交易總金額>>");  
        oPayment.Send.TradeDesc = "<<您該筆訂單的描述>>";  
    }  
}
```

```

oPayment.Send.ChoosePayment = PaymentMethod.CVS; // 或使用 PaymentMethod.BARCODE;
oPayment.Send.Remark = "<<您要填寫的其他備註>>";
oPayment.Send.ChooseSubPayment = PaymentMethodItem.None;
oPayment.Send.NeedExtraPaidInfo = ExtraPaymentInfo.No;
oPayment.Send.DeviceSource = DeviceType.PC;
// 加入選購商品資料。
oPayment.Send.Items.Add(new Item() {
    Name = "<<產品A>>", Price = Decimal.Parse("<<單價>>"), Currency = "<<幣別>>",
    Quantity = Int32.Parse("<<數量>>"), URL = "<<產品說明位址>>"
});
oPayment.Send.Items.Add(new Item() {
    Name = "<<產品B>>", Price = Decimal.Parse("<<單價>>"), Currency = "<<幣別>>",
    Quantity = Int32.Parse("<<數量>>"), URL = "<<產品說明位址>>"
});
oPayment.Send.Items.Add(new Item() {
    Name = "<<產品C>>", Price = Decimal.Parse("<<單價>>"), Currency = "<<幣別>>",
    Quantity = Int32.Parse("<<數量>>"), URL = "<<產品說明位址>>"
});
/* CVS, BARCODE 延伸參數 */
oPayment.SendExtend.Desc_1 = "<<交易描述 1>>";
oPayment.SendExtend.Desc_2 = "<<交易描述 2>>";
oPayment.SendExtend.Desc_3 = "<<交易描述 3>>";
oPayment.SendExtend.Desc_4 = "<<交易描述 4>>";
oPayment.SendExtend.PaymentInfoURL = "<<您要收到付款相關資訊的伺服器端網址>>";
oPayment.SendExtend.ClientRedirectURL = "<<您要收到(超商)條碼相關資訊的伺服器端網址>>";
/* 產生訂單 */
enErrors.AddRange(oPayment.CheckOut());
}
}
catch (Exception ex)
{
    // 例外錯誤處理。
    enErrors.Add(ex.Message);
}
finally
{
    // 顯示錯誤訊息。
    if (enErrors.Count() > 0)

```

```
        string szErrorMessage = String.Join("\r\n", enErrors);  
    }
```

CVS 或 BARCODE 付款方式範例(PHP)

```
/*  
 * 產生訂單的範例程式碼。  
 */  
try  
{  
    $oPayment = new AllInOne();  
    /* 服務參數 */  
    $oPayment->ServiceURL = "<<您要呼叫的服務位址>>";  
    $oPayment->HashKey = "<<AllPay提供給您的Hash Key>>";  
    $oPayment->HashIV = "<<AllPay提供給您的Hash IV>>";  
    $oPayment->MerchantID = "<<AllPay提供給您的特店編號>>";  
    /* 基本參數 */  
    $oPayment->Send['ReturnURL'] = "<<您要收到付款完成通知的伺服器端網址>>";  
    $oPayment->Send['ClientBackURL'] = "<<您要歐付寶返回按鈕導向的瀏覽器端網址>>";  
    $oPayment->Send['OrderResultURL'] = "<<您要收到付款完成通知的瀏覽器端網址>>";  
    $oPayment->Send['MerchantTradeNo'] = "<<您此筆訂單交易編號>>";  
    $oPayment->Send['MerchantTradeDate'] = date('Y/m/d H:i:s');  
    $oPayment->Send['TotalAmount'] = (int) "<<您此筆訂單的交易總金額>>";  
    $oPayment->Send['TradeDesc'] = "<<您該筆訂單的描述>>";  
    $oPayment->Send['ChoosePayment'] = PaymentMethod::CVS; // 或使用 PaymentMethod::BARCODE;  
    $oPayment->Send['Remark'] = "<<您要填寫的其他備註>>";  
    $oPayment->Send['ChooseSubPayment'] = PaymentMethodItem::None;  
    $oPayment->Send['NeedExtraPaidInfo'] = ExtraPaymentInfo::No;  
    $oPayment->Send['DeviceSource'] = DeviceType::PC;  
    // 加入選購商品資料。  
    array_push($oPayment->Send['Items'], array('Name' => "<<產品A>>", 'Price' => (int) "<<單價>>",  
        'Currency' => "<<幣別>>", 'Quantity' => (int) "<<數量>>", 'URL' => "<<產品說明位址>>"));  
    array_push($oPayment->Send['Items'], array('Name' => "<<產品B>>", 'Price' => (int) "<<單價>>",  
        'Currency' => "<<幣別>>", 'Quantity' => (int) "<<數量>>", 'URL' => "<<產品說明位址>>"));  
    array_push($oPayment->Send['Items'], array('Name' => "<<產品C>>", 'Price' => (int) "<<單價>>",  
        'Currency' => "<<幣別>>", 'Quantity' => (int) "<<數量>>", 'URL' => "<<產品說明位址>>"));  
    /* CVS, BARCODE 延伸參數 */  
    $oPayment->SendExtend['Desc_1'] = "<<交易描述 1>>";  
    $oPayment->SendExtend['Desc_2'] = "<<交易描述 2>>";  
}
```

```

$Payment->SendExtend["Desc_3"] = "<<交易描述 3>>";
$Payment->SendExtend["Desc_4"] = "<<交易描述 4>>";
$Payment->SendExtend["PaymentInfoURL"] = "<<您要收到付款相關資訊的伺服器端網址>>";
/* 產生訂單 */
$Payment->CheckOut();
/* 產生產生訂單 Html Code 的方法 */
$szHtml = $Payment->CheckOutString();
}
catch (Exception $e)
{
    // 例外錯誤處理。
    throw $e;
}

```

CVS 或 BARCODE 付款方式範例(JAVA)

```

/*
 * 產生訂單的範例程式碼。
 */
List<String> enErrors = new ArrayList<String>();
try {
    AllInOne oPayment = new AllInOne();

    /* 服務參數 */
    oPayment.ServiceMethod = HttpMethod.HttpPOST;
    oPayment.ServiceURL = "<<您要呼叫的服務位址>>";
    oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";
    oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";
    oPayment.MerchantID = "<<AllPay提供給您的特店編號>>";

    /* 基本參數 */
    oPayment.Send.ReturnURL = "<<您要收到付款完成通知的伺服器端網址>>";
    oPayment.Send.ClientBackURL = "<<您要歐付寶返回按鈕導向的瀏覽器端網址>>";
    oPayment.Send.OrderResultURL = "<<您要收到付款完成通知的瀏覽器端網址>>";
    oPayment.Send.MerchantTradeNo = "<<您此筆訂單交易編號>>";
    oPayment.Send.MerchantTradeDate = new Date(); // "<<您此筆訂單的交易時間>>"
    oPayment.Send.TotalAmount = new Decimal("<<您此筆訂單的交易總金額>>");
    oPayment.Send.TradeDesc = "<<您該筆訂單的描述>>";
    oPayment.Send.ChoosePayment = PaymentMethod.CVS; //或選擇 PaymentMethod.BARCODE;
}

```

```
oPayment.Send.Remark = "<<您要填寫的其他備註>>";
oPayment.Send.ChooseSubPayment = PaymentMethodItem.None;
oPayment.Send.NeedExtraPaidInfo = ExtraPaymentInfo.No;
oPayment.Send.DeviceSource = DeviceType.PC;

// 加入選購商品資料。
Item a1 = new Item();
a1.Name = "<<產品A>>";
a1.Price = new Decimal("<<單價>>");
a1.Currency = "<<幣別>>";
a1.Quantity = 0; // <<數量>>
a1.URL = "<<產品說明位址>>";
oPayment.Send.Items.add(a1);

Item a2 = new Item();
a2.Name = "<<產品B>>";
a2.Price = new Decimal("<<單價>>");
a2.Currency = "<<幣別>>";
a2.Quantity = 0; // <<數量>>
a2.URL = "<<產品說明位址>>";
oPayment.Send.Items.add(a2);

/* CVS, BARCODE 延伸參數 */
oPayment.SendExtend.Desc_1 = "<<交易描述 1>>";
oPayment.SendExtend.Desc_2 = "<<交易描述 2>>";
oPayment.SendExtend.Desc_3 = "<<交易描述 3>>";
oPayment.SendExtend.Desc_4 = "<<交易描述 4>>";
oPayment.SendExtend.PaymentInfoURL = "<<您要收到付款相關資訊的伺服器端網址>>";
oPayment.SendExtend.ClientRedirectURL = "<<您要收到(超商)條碼相關資訊的伺服器端網址>>";

/* 產生訂單 */
enErrors.addAll(oPayment.CheckOut(response.getWriter()));

/* 產生產生訂單 Html Code 的方法 */
StringBuilder szHtml = new StringBuilder();
enErrors.addAll(oPayment.CheckOutString(szHtml));
}

catch (Exception e) {
    // 例外錯誤處理。
}
```

```
enErrors.add(e.getMessage());  
}  
finally {  
    // 顯示錯誤訊息。  
    if (enErrors.size() > 0)  
        out.print(enErrors);  
}
```

4. Alipay 付款方式範例

Alipay 付款方式範例(C#)

```
/*  
 * 產生訂單的範例程式碼。  
 */  
List<string> enErrors = new List<string>();  
  
try  
{  
    using (AllInOne oPayment = new AllInOne())  
    {  
        /* 服務參數 */  
        oPayment.ServiceMethod = HttpMethod.HttpPOST;  
        oPayment.ServiceURL = "<<您要呼叫的服務位址>>";  
        oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";  
        oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";  
        oPayment.MerchantID = "<<AllPay提供給您的特店編號>>";  
        /* 基本參數 */  
        oPayment.Send.ReturnURL = "<<您要收到付款完成通知的伺服器端網址>>";  
        oPayment.Send.ClientBackURL = "<<您要歐付寶返回按鈕導向的瀏覽器端網址>>";  
        oPayment.Send.OrderResultURL = "<<您要收到付款完成通知的瀏覽器端網址>>";  
        oPayment.Send.MerchantTradeNo = "<<您此筆訂單交易編號>>";  
        oPayment.Send.MerchantTradeDate = DateTime.Now("<<您此筆訂單的交易時間>>");  
        oPayment.Send.TotalAmount = Decimal.Parse("<<您此筆訂單的交易總金額>>");  
        oPayment.Send.TradeDesc = "<<您該筆訂單的描述>>";  
        oPayment.Send.ChoosePayment = PaymentMethod.Alipay;  
        oPayment.Send.Remark = "<<您要填寫的其他備註>>";  
        oPayment.Send.ChooseSubPayment = PaymentMethodItem.None;  
        oPayment.Send.NeedExtraPaidInfo = ExtraPaymentInfo.No;
```

```
oPayment.Send.DeviceSource = DeviceType.PC;
// 加入選購商品資料。
oPayment.Send.Items.Add(new Item() {
    Name = "<<產品A>>", Price = Decimal.Parse("<<單價>>"), Currency = "<<幣別>>",
    Quantity = Int32.Parse("<<數量>>"), URL = "<<產品說明位址>>"
});
oPayment.Send.Items.Add(new Item() {
    Name = "<<產品B>>", Price = Decimal.Parse("<<單價>>"), Currency = "<<幣別>>",
    Quantity = Int32.Parse("<<數量>>"), URL = "<<產品說明位址>>"
});
oPayment.Send.Items.Add(new Item() {
    Name = "<<產品C>>", Price = Decimal.Parse("<<單價>>"), Currency = "<<幣別>>",
    Quantity = Int32.Parse("<<數量>>"), URL = "<<產品說明位址>>"
});
/* Alipay 延伸參數 */
oPayment.SendExtend.Email = "<<您的購買人電子郵件信箱>>";
oPayment.SendExtend.PhoneNo = "<<您的購買人連絡電話>>";
oPayment.SendExtend.UserName = "<<您的購買人姓名>>";
/* 產生訂單 */
enErrors.AddRange(oPayment.CheckOut());
/* 產生產生訂單 Html Code 的方法 */
string szHtml = String.Empty;
enErrors.AddRange(oPayment.CheckOutString(ref szHtml));
}
}
catch (Exception ex)
{
    // 例外錯誤處理。
    enErrors.Add(ex.Message);
}
finally
{
    // 顯示錯誤訊息。
    if (enErrors.Count() > 0)
        string szErrorMessage = String.Join("\r\n", enErrors);
}
```

Alipay 付款方式範例(PHP)


```

/*
 * 產生訂單的範例程式碼。
 */
try
{
    $oPayment = new AllInOne();
    /* 服務參數 */
    $oPayment->ServiceURL = "<<您要呼叫的服務位址>>";
    $oPayment->HashKey = "<<AllPay提供給您的Hash Key>>";
    $oPayment->HashIV = "<<AllPay提供給您的Hash IV>>";
    $oPayment->MerchantID = "<<AllPay提供給您的特店編號>>";
    /* 基本參數 */
    $oPayment->Send['ReturnURL'] = "<<您要收到付款完成通知的伺服器端網址>>";
    $oPayment->Send['ClientBackURL'] = "<<您要歐付寶返回按鈕導向的瀏覽器端網址>>";
    $oPayment->Send['OrderResultURL'] = "<<您要收到付款完成通知的瀏覽器端網址>>";
    $oPayment->Send['MerchantTradeNo'] = "<<您此筆訂單交易編號>>";
    $oPayment->Send['MerchantTradeDate'] = date('Y/m/d H:i:s');
    $oPayment->Send['TotalAmount'] = (int) "<<您此筆訂單的交易總金額>>";
    $oPayment->Send['TradeDesc'] = "<<您該筆訂單的描述>>";
    $oPayment->Send['ChoosePayment'] = PaymentMethod::Alipay;
    $oPayment->Send['Remark'] = "<<您要填寫的其他備註>>";
    $oPayment->Send['ChooseSubPayment'] = PaymentMethodItem::None;
    $oPayment->Send['NeedExtraPaidInfo'] = ExtraPaymentInfo::No;
    $oPayment->Send['DeviceSource'] = DeviceType::PC;
    // 加入選購商品資料。
    array_push($oPayment->Send['Items'], array('Name' => "<<產品A>>", 'Price' => (int) "<<單價>>",
        'Currency' => "<<幣別>>", 'Quantity' => (int) "<<數量>>", 'URL' => "<<產品說明位址>>"));
    array_push($oPayment->Send['Items'], array('Name' => "<<產品B>>", 'Price' => (int) "<<單價>>",
        'Currency' => "<<幣別>>", 'Quantity' => (int) "<<數量>>", 'URL' => "<<產品說明位址>>"));
    array_push($oPayment->Send['Items'], array('Name' => "<<產品C>>", 'Price' => (int) "<<單價>>",
        'Currency' => "<<幣別>>", 'Quantity' => (int) "<<數量>>", 'URL' => "<<產品說明位址>>"));
    /* Alipay 延伸參數 */
    $oPayment->SendExtend['Email'] = "<<您的購買人電子郵件信箱>>";
    $oPayment->SendExtend['PhoneNo'] = "<<您的購買人連絡電話>>";
    $oPayment->SendExtend['UserName'] = "<<您的購買人姓名>>";
    /* 產生訂單 */
    $oPayment->Checkout();
    /* 產生產生訂單 Html Code 的方法 */

```

```
$szHtml = $oPayment->CheckOutString();  
}  
catch (Exception $e)  
{  
    // 例外錯誤處理。  
    throw $e;  
}
```

Alipay 付款方式範例(JAVA)

```
/*  
 * 產生訂單的範例程式碼。  
 */  
List<String> enErrors = new ArrayList<String>();  
try {  
    AllInOne oPayment = new AllInOne();  
  
    /* 服務參數 */  
    oPayment.ServiceMethod = HttpMethod.HttpPOST;  
    oPayment.ServiceURL = "<<您要呼叫的服務位址>>";  
    oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";  
    oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";  
    oPayment.MerchantID = "<<AllPay提供給您的特店編號>>";  
  
    /* 基本參數 */  
    oPayment.Send.ReturnURL = "<<您要收到付款完成通知的伺服器端網址>>";  
    oPayment.Send.ClientBackURL = "<<您要歐付寶返回按鈕導向的瀏覽器端網址>>";  
    oPayment.Send.OrderResultURL = "<<您要收到付款完成通知的瀏覽器端網址>>";  
    oPayment.Send.MerchantTradeNo = "<<您此筆訂單交易編號>>";  
    oPayment.Send.MerchantTradeDate = new Date();/"<<您此筆訂單的交易時間>>"  
    oPayment.Send.TotalAmount = new Decimal("<<您此筆訂單的交易總金額>>");  
    oPayment.Send.TradeDesc = "<<您該筆訂單的描述>>";  
    oPayment.Send.ChoosePayment = PaymentMethod.Alipay;  
    oPayment.Send.Remark = "<<您要填寫的其他備註>>";  
    oPayment.Send.ChooseSubPayment = PaymentMethodItem.None;  
    oPayment.Send.NeedExtraPaidInfo = ExtraPaymentInfo.No;  
    oPayment.Send.DeviceSource = DeviceType.PC;  
  
    // 加入選購商品資料。
```

```
Item a1 = new Item();
a1.Name = "<<產品A>>";
a1.Price = new Decimal("<<單價>>");
a1.Currency = "<<幣別>>";
a1.Quantity = 0;//<<數量>>
a1.URL = "<<產品說明位址>>";
oPayment.Send.Items.add(a1);

Item a2 = new Item();
a2.Name = "<<產品B>>";
a2.Price = new Decimal("<<單價>>");
a2.Currency = "<<幣別>>";
a2.Quantity = 0;//<<數量>>
a2.URL = "<<產品說明位址>>";
oPayment.Send.Items.add(a2);

/* Alipay 延伸參數 */
oPayment.SendExtend.Email = "<<您的購買人電子郵件信箱>>";
oPayment.SendExtend.PhoneNo = "<<您的購買人連絡電話>>";
oPayment.SendExtend.UserName = "<<您的購買人姓名>>";
/* 產生訂單 */
enErrors.addAll(oPayment.CheckOut(response.getWriter()));
/* 產生產生訂單 Html Code 的方法 */
StringBuilder szHtml = new StringBuilder();
enErrors.addAll(oPayment.CheckOutString(szHtml));
}
catch (Exception e) {
    // 例外錯誤處理。
    enErrors.add(e.getMessage());
}
finally {
    // 顯示錯誤訊息。
    if (enErrors.size() > 0)
        out.print(enErrors);
}
```

5. Tenpay 付款方式範例

Tenpay 付款方式範例(C#)

```
/*
 * 產生訂單的範例程式碼。
 */
List<string> enErrors = new List<string>();

try
{
    using (AllInOne oPayment = new AllInOne())
    {
        /* 服務參數 */
        oPayment.ServiceMethod = HttpMethod.HttpPOST;
        oPayment.ServiceURL = "<<您要呼叫的服務位址>>";
        oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";
        oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";
        oPayment.MerchantID = "<<AllPay提供給您的特店編號>>";
        /* 基本參數 */
        oPayment.Send.ReturnURL = "<<您要收到付款完成通知的伺服器端網址>>";
        oPayment.Send.ClientBackURL = "<<您要歐付寶返回按鈕導向的瀏覽器端網址>>";
        oPayment.Send.OrderResultURL = "<<您要收到付款完成通知的瀏覽器端網址>>";
        oPayment.Send.MerchantTradeNo = "<<您此筆訂單交易編號>>";
        oPayment.Send.MerchantTradeDate = DateTime.Now("<<您此筆訂單的交易時間>>");
        oPayment.Send.TotalAmount = Decimal.Parse("<<您此筆訂單的交易總金額>>");
        oPayment.Send.TradeDesc = "<<您該筆訂單的描述>>";
        oPayment.Send.ChoosePayment = PaymentMethod.Tenpay;
        oPayment.Send.Remark = "<<您要填寫的其他備註>>";
        oPayment.Send.ChooseSubPayment = PaymentMethodItem.None;
        oPayment.Send.NeedExtraPaidInfo = ExtraPaymentInfo.No;
        oPayment.Send.DeviceSource = DeviceType.PC;
        // 加入選購商品資料。
        oPayment.Send.Items.Add(new Item() {
            Name = "<<產品A>>", Price = Decimal.Parse("<<單價>>"), Currency = "<<幣別>>",
            Quantity = Int32.Parse("<<數量>>"), URL = "<<產品說明位址>>"
        });
        oPayment.Send.Items.Add(new Item() {
            Name = "<<產品B>>", Price = Decimal.Parse("<<單價>>"), Currency = "<<幣別>>",
```

```

        Quantity = Int32.Parse("<<數量>>"), URL = "<<產品說明位址>>"
    });
    oPayment.Send.Items.Add(new Item() {
        Name = "<<產品C>>", Price = Decimal.Parse("<<單價>>"), Currency = "<<幣別>>",
        Quantity = Int32.Parse("<<數量>>"), URL = "<<產品說明位址>>"
    });
    /* Tenpay 延伸參數 */
    oPayment.Send.Extend.ExpireTime = DateTime.Parse("<<您的付款截止時間>>");
    /* 產生訂單 */
    enErrors.AddRange(oPayment.CheckOut());
    /* 產生產生訂單 Html Code 的方法 */
    string szHtml = String.Empty;
    enErrors.AddRange(oPayment.CheckOutString(ref szHtml));
}
}
catch (Exception ex)
{
    // 例外錯誤處理。
    enErrors.Add(ex.Message);
}
finally
{
    // 顯示錯誤訊息。
    if (enErrors.Count() > 0)
        string szErrorMessage = String.Join("\r\n", enErrors);
}
}

```

Tenpay 付款方式範例(PHP)

```

/*
 * 產生訂單的範例程式碼。
 */
try
{
    $oPayment = new AllInOne();
    /* 服務參數 */
    $oPayment->ServiceURL = "<<您要呼叫的服務位址>>";
    $oPayment->HashKey = "<<AllPay提供給您的Hash Key>>";
    $oPayment->HashIV = "<<AllPay提供給您的Hash IV>>";
}

```

```

$Payment->MerchantID = "<<AllPay提供給您的特店編號>>";
/* 基本參數 */
$Payment->Send['ReturnURL'] = "<<您要收到付款完成通知的伺服器端網址>>";
$Payment->Send['ClientBackURL'] = "<<您要歐付寶返回按鈕導向的瀏覽器端網址>>";
$Payment->Send['OrderResultURL'] = "<<您要收到付款完成通知的瀏覽器端網址>>";
$Payment->Send['MerchantTradeNo'] = "<<您此筆訂單交易編號>>";
$Payment->Send['MerchantTradeDate'] = date("Y/m/d H:i:s");
$Payment->Send['TotalAmount'] = (int) "<<您此筆訂單的交易總金額>>";
$Payment->Send['TradeDesc'] = "<<您該筆訂單的描述>>";
$Payment->Send['ChoosePayment'] = PaymentMethod::Tenpay;
$Payment->Send['Remark'] = "<<您要填寫的其他備註>>";
$Payment->Send['ChooseSubPayment'] = PaymentMethodItem::None;
$Payment->Send['NeedExtraPaidInfo'] = ExtraPaymentInfo::No;
$Payment->Send['DeviceSource'] = DeviceType::PC;
// 加入選購商品資料。
array_push($Payment->Send['Items'], array('Name' => "<<產品A>>", 'Price' => (int)"<<單價>>",
    'Currency' => "<<幣別>>", 'Quantity' => (int) "<<數量>>", 'URL' => "<<產品說明位址>>"));
array_push($Payment->Send['Items'], array('Name' => "<<產品B>>", 'Price' => (int)"<<單價>>",
    'Currency' => "<<幣別>>", 'Quantity' => (int) "<<數量>>", 'URL' => "<<產品說明位址>>"));
array_push($Payment->Send['Items'], array('Name' => "<<產品C>>", 'Price' => (int)"<<單價>>",
    'Currency' => "<<幣別>>", 'Quantity' => (int) "<<數量>>", 'URL' => "<<產品說明位址>>"));
/* Tenpay 延伸參數 */
$Payment->SendExtend['ExpireTime'] = date("Y/m/d H:i:s", mktime(date("H"), date("i"), date("s"), date("m"),
date("d") + 3, date("Y")));
/* 產生訂單 */
$Payment->Checkout();
/* 產生產生訂單 Html Code 的方法 */
$szHtml = $Payment->CheckoutString();
}
catch (Exception $e)
{
    // 例外錯誤處理。
    throw $e;
}
    
```

Tenpay 付款方式範例(JAVA)

```

/*
 * 產生訂單的範例程式碼。
    
```

```
*/  
List<String> enErrors = new ArrayList<String>();  
try {  
    AllInOne oPayment = new AllInOne();  
  
    /* 服務參數 */  
    oPayment.ServiceMethod = HttpMethod.HttpPOST;  
    oPayment.ServiceURL = "<<您要呼叫的服務位址>>";  
    oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";  
    oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";  
    oPayment.MerchantID = "<<AllPay提供給您的特店編號>>";  
  
    /* 基本參數 */  
    oPayment.Send.ReturnURL = "<<您要收到付款完成通知的伺服器端網址>>";  
    oPayment.Send.ClientBackURL = "<<您要歐付寶返回按鈕導向的瀏覽器端網址>>";  
    oPayment.Send.OrderResultURL = "<<您要收到付款完成通知的瀏覽器端網址>>";  
    oPayment.Send.MerchantTradeNo = "<<您此筆訂單交易編號>>";  
    oPayment.Send.MerchantTradeDate = new Date();/"<<您此筆訂單的交易時間>>"  
    oPayment.Send.TotalAmount = new Decimal("<<您此筆訂單的交易總金額>>");  
    oPayment.Send.TradeDesc = "<<您該筆訂單的描述>>";  
    oPayment.Send.ChoosePayment = PaymentMethod.Alipay;  
    oPayment.Send.Remark = "<<您要填寫的其他備註>>";  
    oPayment.Send.ChooseSubPayment = PaymentMethodItem.None;  
    oPayment.Send.NeedExtraPaidInfo = ExtraPaymentInfo.No;  
    oPayment.Send.DeviceSource = DeviceType.PC;  
  
    // 加入選購商品資料。  
    Item a1 = new Item();  
    a1.Name = "<<產品A>>";  
    a1.Price = new Decimal("<<單價>>");  
    a1.Currency = "<<幣別>>";  
    a1.Quantity = 0;"<<數量>>  
    a1.URL = "<<產品說明位址>>";  
    oPayment.Send.Items.add(a1);  
  
    Item a2 = new Item();  
    a2.Name = "<<產品B>>";  
    a2.Price = new Decimal("<<單價>>");
```

```

        a2.Currency = "<<幣別>>";
        a2.Quantity = 0;//<<數量>>
        a2.URL = "<<產品說明位址>>";
        oPayment.Send.Items.add(a2);

        /* Tenpay 延伸參數 */
        oPayment.SendExtend.ExpireTime = new Date();//"<<您的付款截止時間>>";
        /* 產生訂單 */
        enErrors.addAll(oPayment.CheckOut(response.getWriter()));
        /* 產生產生訂單 Html Code 的方法 */
        StringBuilder szHtml = new StringBuilder();
        enErrors.addAll(oPayment.CheckOutString(szHtml));
    }
    catch (Exception e) {
        // 例外錯誤處理。
        enErrors.add(e.getMessage());
    }
    finally {
        // 顯示錯誤訊息。
        if (enErrors.size() > 0)
            out.print(enErrors);
    }
}

```

6. Credit 分期付款方式範例

Credit 分期付款方式範例(C#)

```

/*
 * 產生訂單的範例程式碼。
 */
List<string> enErrors = new List<string>();

try
{
    using (AllInOne oPayment = new AllInOne())
    {
        /* 服務參數 */
        oPayment.ServiceMethod = HttpMethod.HttpPOST;
    }
}

```



```
oPayment.ServiceURL = "<<您要呼叫的服務位址>>";
oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";
oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";
oPayment.MerchantID = "<<AllPay提供給您的特店編號>>";
/* 基本參數 */
oPayment.Send.ReturnURL = "<<您要收到付款完成通知的伺服器端網址>>";
oPayment.Send.ClientBackURL = "<<您要歐付寶返回按鈕導向的瀏覽器端網址>>";
oPayment.Send.OrderResultURL = "<<您要收到付款完成通知的瀏覽器端網址>>";
oPayment.Send.MerchantTradeNo = "<<您此筆訂單交易編號>>";
oPayment.Send.MerchantTradeDate = DateTime.Now("<<您此筆訂單的交易時間>>");
oPayment.Send.TotalAmount = Decimal.Parse("<<您此筆訂單的交易總金額>>");
oPayment.Send.TradeDesc = "<<您該筆訂單的描述>>";
oPayment.Send.ChoosePayment = PaymentMethod.Credit;
oPayment.Send.Remark = "<<您要填寫的其他備註>>";
oPayment.Send.ChooseSubPayment = PaymentMethodItem.None;
oPayment.Send.NeedExtraPaidInfo = ExtraPaymentInfo.No;
oPayment.Send.DeviceSource = DeviceType.PC;
// 加入選購商品資料。
oPayment.Send.Items.Add(new Item() {
    Name = "<<產品A>>", Price = Decimal.Parse("<<單價>>"), Currency = "<<幣別>>",
    Quantity = Int32.Parse("<<數量>>"), URL = "<<產品說明位址>>"
});
oPayment.Send.Items.Add(new Item() {
    Name = "<<產品B>>", Price = Decimal.Parse("<<單價>>"), Currency = "<<幣別>>",
    Quantity = Int32.Parse("<<數量>>"), URL = "<<產品說明位址>>"
});
oPayment.Send.Items.Add(new Item() {
    Name = "<<產品C>>", Price = Decimal.Parse("<<單價>>"), Currency = "<<幣別>>",
    Quantity = Int32.Parse("<<數量>>"), URL = "<<產品說明位址>>"
});
/* Credit 分期延伸參數 */
oPayment.SendExtend.CreditInstallment = Int32.Parse("<<您的刷卡分期數>>");
oPayment.SendExtend.InstallmentAmount = Decimal.Parse("<<您使用刷卡分期的付款金額>>");
oPayment.SendExtend.Redeem = Boolean.Parse("<<是否使用紅利折抵>>");
oPayment.SendExtend.UnionPay = Boolean.Parse("<<是否為聯營卡>>");
/* 產生訂單 */
enErrors.AddRange(oPayment.CheckOut());
/* 產生產生訂單 Html Code 的方法 */
```

```
        string szHtml = String.Empty;
        enErrors.AddRange(oPayment.CheckOutString(ref szHtml));
    }
}
catch (Exception ex)
{
    // 例外錯誤處理。
    enErrors.Add(ex.Message);
}
finally
{
    // 顯示錯誤訊息。
    if (enErrors.Count() > 0)
        string szErrorMessage = String.Join("\r\n", enErrors);
}
```

Credit 分期付款方式範例(PHP)

```
/*
 * 產生訂單的範例程式碼。
 */
try
{
    $oPayment = new AllInOne();
    /* 服務參數 */
    $oPayment->ServiceURL = "<<您要呼叫的服務位址>>";
    $oPayment->HashKey = "<<AllPay提供給您的Hash Key>>";
    $oPayment->HashIV = "<<AllPay提供給您的Hash IV>>";
    $oPayment->MerchantID = "<<AllPay提供給您的特店編號>>";
    /* 基本參數 */
    $oPayment->Send['ReturnURL'] = "<<您要收到付款完成通知的伺服器端網址>>";
    $oPayment->Send['ClientBackURL'] = "<<您要歐付寶返回按鈕導向的瀏覽器端網址>>";
    $oPayment->Send['OrderResultURL'] = "<<您要收到付款完成通知的瀏覽器端網址>>";
    $oPayment->Send['MerchantTradeNo'] = "<<您此筆訂單交易編號>>";
    $oPayment->Send['MerchantTradeDate'] = date('Y/m/d H:i:s');
    $oPayment->Send['TotalAmount'] = (int) "<<您此筆訂單的交易總金額>>";
    $oPayment->Send['TradeDesc'] = "<<您該筆訂單的描述>>";
    $oPayment->Send['ChoosePayment'] = PaymentMethod::Credit;
    $oPayment->Send['Remark'] = "<<您要填寫的其他備註>>";
}
```

```

$Payment->Send['ChooseSubPayment'] = PaymentMethodItem::None;
$Payment->Send['NeedExtraPaidInfo'] = ExtraPaymentInfo::No;
$Payment->Send['DeviceSource'] = DeviceType::PC;
// 加入選購商品資料。
array_push($Payment->Send['Items'], array('Name' => "<<產品A>>", 'Price' => (int)"<<單價>>",
    'Currency' => "<<幣別>>", 'Quantity' => (int) "<<數量>>", 'URL' => "<<產品說明位址>>"));
array_push($Payment->Send['Items'], array('Name' => "<<產品B>>", 'Price' => (int)"<<單價>>",
    'Currency' => "<<幣別>>", 'Quantity' => (int) "<<數量>>", 'URL' => "<<產品說明位址>>"));
array_push($Payment->Send['Items'], array('Name' => "<<產品C>>", 'Price' => (int)"<<單價>>",
    'Currency' => "<<幣別>>", 'Quantity' => (int) "<<數量>>", 'URL' => "<<產品說明位址>>"));
/* Credit 分期延伸參數 */
$Payment->SendExtend['CreditInstallment'] = (int) "<<您的刷卡分期數>>";
$Payment->SendExtend['InstallmentAmount'] = (int) "<<您使用刷卡分期的付款金額>>";
$Payment->SendExtend['Redeem'] = (bool)"<<是否使用紅利折抵>>";
$Payment->SendExtend['UnionPay'] = (bool) "<<是否為聯營卡>>";
/* 產生訂單 */
$Payment->Checkout();
/* 產生產生訂單 Html Code 的方法 */
$szHtml = $Payment->CheckoutString();
}
catch (Exception $e)
{
    // 例外錯誤處理。
    throw $e;
}
    
```

Credit 分期付款方式範例(JAVA)

```

/*
 * 產生訂單的範例程式碼。
 */
List<String> enErrors = new ArrayList<String>();
try {
    AllInOne oPayment = new AllInOne();

    /* 服務參數 */
    oPayment.ServiceMethod = HttpMethod.HttpPOST;
    oPayment.ServiceURL = "<<您要呼叫的服務位址>>";
    oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";
    
```

```
oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";
oPayment.MerchantID = "<<AllPay提供給您的特店編號>>";

/* 基本參數 */
oPayment.Send.ReturnURL = "<<您要收到付款完成通知的伺服器端網址>>";
oPayment.Send.ClientBackURL = "<<您要歐付寶返回按鈕導向的瀏覽器端網址>>";
oPayment.Send.OrderResultURL = "<<您要收到付款完成通知的瀏覽器端網址>>";
oPayment.Send.MerchantTradeNo = "<<您此筆訂單交易編號>>";
oPayment.Send.MerchantTradeDate = new Date();/"<<您此筆訂單的交易時間>>"
oPayment.Send.TotalAmount = new Decimal("<<您此筆訂單的交易總金額>>");
oPayment.Send.TradeDesc = "<<您該筆訂單的描述>>";
oPayment.Send.ChoosePayment = PaymentMethod.Alipay;
oPayment.Send.Remark = "<<您要填寫的其他備註>>";
oPayment.Send.ChooseSubPayment = PaymentMethodItem.None;
oPayment.Send.NeedExtraPaidInfo = ExtraPaymentInfo.No;
oPayment.Send.DeviceSource = DeviceType.PC;

// 加入選購商品資料。
Item a1 = new Item();
a1.Name = "<<產品A>>";
a1.Price = new Decimal("<<單價>>");
a1.Currency = "<<幣別>>";
a1.Quantity = 0;/"<<數量>>"
a1.URL = "<<產品說明位址>>";
oPayment.Send.Items.add(a1);

Item a2 = new Item();
a2.Name = "<<產品B>>";
a2.Price = new Decimal("<<單價>>");
a2.Currency = "<<幣別>>";
a2.Quantity = 0;/"<<數量>>"
a2.URL = "<<產品說明位址>>";
oPayment.Send.Items.add(a2);

/* Credit 分期延伸參數 */
oPayment.SendExtend.CreditInstallment = 0;/"<<您的刷卡分期數>>";
oPayment.SendExtend.InstallmentAmount = new Decimal("<<您使用刷卡分期的付款金額>>");
oPayment.SendExtend.Redeem = false;/"<<是否使用紅利折抵>>";
```

```
oPayment.SendExtend.UnionPay = false; /* "<<是否為聯營卡>>";  
/* 產生訂單 */  
enErrors.addAll(oPayment.CheckOut(response.getWriter()));  
/* 產生產生訂單 Html Code 的方法 */  
StringBuilder szHtml = new StringBuilder();  
enErrors.addAll(oPayment.CheckOutString(szHtml));  
  
}  
catch (Exception e) {  
    // 例外錯誤處理。  
    enErrors.add(e.getMessage());  
}  
finally {  
    // 顯示錯誤訊息。  
    if (enErrors.size() > 0)  
        out.print(enErrors);  
}
```

7. Credit 定期定額付款方式範例

Credit 定期定額付款方式範例(C#)

```
/*  
 * 產生訂單的範例程式碼。  
 */  
List<string> enErrors = new List<string>();  
  
try  
{  
    using (AllInOne oPayment = new AllInOne())  
    {  
        /* 服務參數 */  
        oPayment.ServiceMethod = HttpMethod.HttpPOST;  
        oPayment.ServiceURL = "<<您要呼叫的服務位址>>";  
        oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";  
        oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";  
        oPayment.MerchantID = "<<AllPay提供給您的特店編號>>";  
        /* 基本參數 */  
        oPayment.Send.ReturnURL = "<<您要收到付款完成通知的伺服器端網址>>";  
    }  
}
```

```

oPayment.Send.ClientBackURL = "<<您要歐付寶返回按鈕導向的瀏覽器端網址>>";
oPayment.Send.OrderResultURL = "<<您要收到付款完成通知的瀏覽器端網址>>";
oPayment.Send.MerchantTradeNo = "<<您此筆訂單交易編號>>";
oPayment.Send.MerchantTradeDate = DateTime.Now("<<您此筆訂單的交易時間>>");
oPayment.Send.TotalAmount = Decimal.Parse("<<您此筆訂單的交易總金額>>");
oPayment.Send.TradeDesc = "<<您該筆訂單的描述>>";
oPayment.Send.ChoosePayment = PaymentMethod.Credit;
oPayment.Send.Remark = "<<您要填寫的其他備註>>";
oPayment.Send.ChooseSubPayment = PaymentMethodItem.None;
oPayment.Send.NeedExtraPaidInfo = ExtraPaymentInfo.No;
oPayment.Send.DeviceSource = DeviceType.PC;
// 加入選購商品資料。
oPayment.Send.Items.Add(new Item() {
    Name = "<<產品A>>", Price = Decimal.Parse("<<單價>>"), Currency = "<<幣別>>",
    Quantity = Int32.Parse("<<數量>>"), URL = "<<產品說明位址>>"
});
oPayment.Send.Items.Add(new Item() {
    Name = "<<產品B>>", Price = Decimal.Parse("<<單價>>"), Currency = "<<幣別>>",
    Quantity = Int32.Parse("<<數量>>"), URL = "<<產品說明位址>>"
});
oPayment.Send.Items.Add(new Item() {
    Name = "<<產品C>>", Price = Decimal.Parse("<<單價>>"), Currency = "<<幣別>>",
    Quantity = Int32.Parse("<<數量>>"), URL = "<<產品說明位址>>"
});
/* Credit 定期定額延伸參數 */
oPayment.SendExtend.PeriodAmount = Int32.Parse("<<每次授權金額>>");
oPayment.SendExtend.PeriodType = PeriodType.Month;
oPayment.SendExtend.Frequency = Int32.Parse("<<執行頻率>>");
oPayment.SendExtend.ExecTimes = Int32.Parse("<<執行次數>>");
oPayment.SendExtend.PeriodReturnURL = "<<定期定額執行結果的回應位址>>";
/* 產生訂單 */
enErrors.AddRange(oPayment.CheckOut());
/* 產生產生訂單 Html Code 的方法 */
string szHtml = String.Empty;
enErrors.AddRange(oPayment.CheckOutString(ref szHtml));
}
}

```

```
catch (Exception ex)
```

```

{
    // 例外錯誤處理。
    enErrors.Add(ex.Message);
}
finally
{
    // 顯示錯誤訊息。
    if (enErrors.Count() > 0)
        string szErrorMessage = String.Join("\r\n", enErrors);
}

```

Credit 定期定額付款方式範例(PHP)

```

/*
 * 產生訂單的範例程式碼。
 */
try
{
    $oPayment = new AllInOne();
    /* 服務參數 */
    $oPayment->ServiceURL = "<<您要呼叫的服務位址>>";
    $oPayment->HashKey = "<<AllPay提供給您的Hash Key>>";
    $oPayment->HashIV = "<<AllPay提供給您的Hash IV>>";
    $oPayment->MerchantID = "<<AllPay提供給您的特店編號>>";
    /* 基本參數 */
    $oPayment->Send['ReturnURL'] = "<<您要收到付款完成通知的伺服器端網址>>";
    $oPayment->Send['ClientBackURL'] = "<<您要歐付寶返回按鈕導向的瀏覽器端網址>>";
    $oPayment->Send['OrderResultURL'] = "<<您要收到付款完成通知的瀏覽器端網址>>";
    $oPayment->Send['MerchantTradeNo'] = "<<您此筆訂單交易編號>>";
    $oPayment->Send['MerchantTradeDate'] = date('Y/m/d H:i:s');
    $oPayment->Send['TotalAmount'] = (int) "<<您此筆訂單的交易總金額>>";
    $oPayment->Send['TradeDesc'] = "<<您該筆訂單的描述>>";
    $oPayment->Send['ChoosePayment'] = PaymentMethod::Credit;
    $oPayment->Send['Remark'] = "<<您要填寫的其他備註>>";
    $oPayment->Send['ChooseSubPayment'] = PaymentMethodItem::None;
    $oPayment->Send['NeedExtraPaidInfo'] = ExtraPaymentInfo::No;
    $oPayment->Send['DeviceSource'] = DeviceType::PC;
    // 加入選購商品資料。
    array_push($oPayment->Send['Items'], array('Name' => "<<產品A>>", 'Price' => (int)"<<單價>>",

```

```

        'Currency' => "<<幣別>>", 'Quantity' => (int) "<<數量>>", 'URL' => "<<產品說明位址>>"));
    array_push($oPayment->Send['Items'], array('Name' => "<<產品B>>", 'Price' => (int) "<<單價>>",
        'Currency' => "<<幣別>>", 'Quantity' => (int) "<<數量>>", 'URL' => "<<產品說明位址>>"));
    array_push($oPayment->Send['Items'], array('Name' => "<<產品C>>", 'Price' => (int) "<<單價>>",
        'Currency' => "<<幣別>>", 'Quantity' => (int) "<<數量>>", 'URL' => "<<產品說明位址>>"));
    /* Credit 定期定額延伸參數 */
    $oPayment->SendExtend['PeriodAmount'] = (int) "<<每次授權金額>>";
    $oPayment->SendExtend['PeriodType'] = PeriodType::Month;
    $oPayment->SendExtend['Frequency'] = (int) "<<執行頻率>>";
    $oPayment->SendExtend['ExecTimes'] = (int) "<<執行次數>>";
    $oPayment->SendExtend['PeriodReturnURL'] = "<<定期定額執行結果的回應位址>>";
    /* 產生訂單 */
    $oPayment->Checkout();
    /* 產生產生訂單 Html Code 的方法 */
    $szHtml = $oPayment->CheckoutString();
}
catch (Exception $e)
{
    // 例外錯誤處理。
    throw $e;
}

```

Credit 定期定額付款方式範例(JAVA)

```

/*
 * 產生訂單的範例程式碼。
 */
List<String> enErrors = new ArrayList<String>();
try {
    AllInOne oPayment = new AllInOne();

    /* 服務參數 */
    oPayment.ServiceMethod = HttpMethod.HttpPOST;
    oPayment.ServiceURL = "<<您要呼叫的服務位址>>";
    oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";
    oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";
    oPayment.MerchantID = "<<AllPay提供給您的特店編號>>";

    /* 基本參數 */

```



```
oPayment.Send.ReturnURL = "<<您要收到付款完成通知的伺服器端網址>>";
oPayment.Send.ClientBackURL = "<<您要歐付寶返回按鈕導向的瀏覽器端網址>>";
oPayment.Send.OrderResultURL = "<<您要收到付款完成通知的瀏覽器端網址>>";
oPayment.Send.MerchantTradeNo = "<<您此筆訂單交易編號>>";
oPayment.Send.MerchantTradeDate = new Date();/"<<您此筆訂單的交易時間>>"
oPayment.Send.TotalAmount = new Decimal("<<您此筆訂單的交易總金額>>");
oPayment.Send.TradeDesc = "<<您該筆訂單的描述>>";
oPayment.Send.ChoosePayment = PaymentMethod.Alipay;
oPayment.Send.Remark = "<<您要填寫的其他備註>>";
oPayment.Send.ChooseSubPayment = PaymentMethodItem.None;
oPayment.Send.NeedExtraPaidInfo = ExtraPaymentInfo.No;
oPayment.Send.DeviceSource = DeviceType.PC;
```

```
// 加入選購商品資料。
```

```
Item a1 = new Item();
a1.Name = "<<產品A>>";
a1.Price = new Decimal("<<單價>>");
a1.Currency = "<<幣別>>";
a1.Quantity = 0;/"<<數量>>"
a1.URL = "<<產品說明位址>>";
oPayment.Send.Items.add(a1);
```

```
Item a2 = new Item();
a2.Name = "<<產品B>>";
a2.Price = new Decimal("<<單價>>");
a2.Currency = "<<幣別>>";
a2.Quantity = 0;/"<<數量>>"
a2.URL = "<<產品說明位址>>";
oPayment.Send.Items.add(a2);
```

```
/* Credit 定期定額延伸參數 */
```

```
oPayment.SendExtend.PeriodAmount = new Decimal("<<每次授權金額>>");
oPayment.SendExtend.PeriodType = PeriodType.Month;
oPayment.SendExtend.Frequency = 0;/"<<執行頻率>>";
oPayment.SendExtend.ExecTimes = 0;/"<<執行次數>>";
oPayment.SendExtend.PeriodReturnURL = "<<定期定額執行結果的回應位址>>";
```

```
/* 產生訂單 */
```

```
enErrors.addAll(oPayment.CheckOut(response.getWriter()));
```

```

/* 產生產生訂單 Html Code 的方法 */

StringBuilder szHtml = new StringBuilder();

enErrors.addAll(oPayment.CheckOutString(szHtml));

}

catch (Exception e) {
    // 例外錯誤處理。

    enErrors.add(e.getMessage());
}

finally {
    // 顯示錯誤訊息。

    if (enErrors.size() > 0)
        out.print(enErrors);
}

```

5.1.2. 取得付款結果通知的方法(CheckOutFeedback)

當 AllPay 會員付款後，或是訂單建立成功後，AllPay 會以 Server POST 的方式傳送處理結果至產生訂單時指定的位址(註 1)，請於收到訊息後，回應已接收訊息。

回應訊息格式為：

成功：1|OK

錯誤：0|<<您的錯誤訊息內容>>

※注意事項:

註 1：指定的位址有 ReturnURL、PaymentInfoURL、PeriodReturnURL 與 OrderResultURL 四種。

(1) 當賣家會員收到 Server POST 訊息後，除 OrderResultURL 指定的位址外，其餘請直接回應訊息在網頁上。

(2) 若沒有收到正確消息，系統會隔三分鐘後重發訊息給特店(賣家會員)，三次後則會延至隔天再發訊息。

(3) 除了回應訊息外，請勿有任何 HTML 標籤、空格等其它符號。

5.1.2.1. 一般付款結果通知參數與範例

當 AllPay 會員付款後，或是貨到付款的訂單建立成功後，參數會以 Server POST 方式傳送至您設定的 ReturnURL 位址。

接收一般付款結果通知的參數成員如下表：

參數	名稱	型態	說明
 MerchantID	廠商編號	String(10)	AllPay 提供的廠商編號。
 MerchantTradeNo	廠商交易編號	String(20)	產生訂單時傳送給 AllPay 的廠商交易編號。
 RtnCode	交易狀態	Integer	1 表示付款成功，800 表示貨到付款訂單建立成功，其餘代碼為失敗(請參考 交易訊息代碼一覽表)。
 RtnMsg	交易訊息	String(200)	
 TradeNo	AllPay 的交易編號	String(20)	請保存 AllPay 的交易編號與 MerchantTradeNo 的關連。
 TradeAmt	交易金額	Decimal	若歐付寶會員選擇信用卡分期付款時，此交易金額會回傳商家在訂單產生時，InstallmentAmount 所帶的金額。
 PaymentDate	付款時間	DateTime	格式為 yyyy/MM/dd HH:mm:ss。
 PaymentType	會員選擇的付款方式	PaymentMethodItem	
 PaymentTypeChargeFee	通路費	Decimal	
 TradeDate	訂單成立時間	DateTime	格式為 yyyy/MM/dd HH:mm:ss。
 SimulatePaid	是否為模擬付款	Integer	1：模擬付款 0：非模擬付款 賣家會員可透過廠商後台網站的履約保證訂單管理來針對單筆訂單模擬由歐付寶發動付款通知，以方便介接 API 的測試(註 1)。

※注意事項：

註 1：當 SimulatePaid 的值為 1 時，RtnCode 也會是 1，表示此筆訂單是模擬付款，並非是由消費者實際真的付款，所以歐付寶也不會撥款給廠商，請勿對該筆交易做出貨等動作，以避免損失。

一般付款結果通知範例(C#)：

```

/*
 * 接收訂單資料產生完成的範例程式碼。
 */
List<string> enErrors = new List<string>();
Hashtable htFeedback = null;
    
```

```

try
{
    using (AllInOne oPayment = new AllInOne())
    {
        oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";
        oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";
        /* 取回付款結果 */
        enErrors.AddRange(oPayment.CheckOutFeedback(ref htFeedback));
    }
    // 取回所有資料
    if (enErrors.Count() == 0)
    {
        /* 支付後的回傳的基本參數 */
        string szMerchantID = String.Empty;
        string szMerchantTradeNo = String.Empty;
        string szPaymentDate = String.Empty;
        string szPaymentType = String.Empty;
        string szPaymentTypeChargeFee = String.Empty;
        string szRtnCode = String.Empty;
        string szRtnMsg = String.Empty;
        string szSimulatePaid = String.Empty;
        string szTradeAmt = String.Empty;
        string szTradeDate = String.Empty;
        string szTradeNo = String.Empty;
        // 取得資料
        foreach (string szKey in htFeedback.Keys)
        {
            switch (szKey)
            {
                /* 支付後的回傳的基本參數 */
                case "MerchantID": szMerchantID = this.Request.Form[szKey]; break;
                case "MerchantTradeNo": szMerchantTradeNo = this.Request.Form[szKey]; break;
                case "PaymentDate": szPaymentDate = this.Request.Form[szKey]; break;
                case "PaymentType": szPaymentType = this.Request.Form[szKey]; break;
                case "PaymentTypeChargeFee": szPaymentTypeChargeFee = this.Request.Form[szKey]; break;
                case "RtnCode": szRtnCode = this.Request.Form[szKey]; break;
                case "RtnMsg": szRtnMsg = this.Request.Form[szKey]; break;
            }
        }
    }
}

```

```

        case "SimulatePaid": szSimulatePaid = this.Request.Form[szKey]; break;
        case "TradeAmt": szTradeAmt = this.Request.Form[szKey]; break;
        case "TradeDate": szTradeDate = this.Request.Form[szKey]; break;
        case "TradeNo": szTradeNo = this.Request.Form[szKey]; break;
        default: break;
    }
}
// 其他資料處理。
.....
} else {
    // 其他資料處理。
    .....
}
}
catch (Exception ex)
{
    // 例外錯誤處理。
    enErrors.Add(ex.Message);
}
finally
{
    this.Response.Clear();
    // 回覆成功訊息。
    if (enErrors.Count() == 0)
        this.Response.Write("1|OK");
    // 回覆錯誤訊息。
    else
        this.Response.Write(String.Format("0|{0}", String.Join("\r\n", enErrors)));
    this.Response.Flush();
    this.Response.End();
}

```

一般付款結果通知範例(PHP)：

```

/*
 * 接收訂單資料產生完成的範例程式碼。
 */
try
{

```

```
$oPayment = new AllInOne();  
/* 服務參數 */  
$oPayment->HashKey = "<<AllPay提供給您的Hash Key>>";  
$oPayment->HashIV = "<<AllPay提供給您的Hash IV>>";  
$oPayment->MerchantID = "<<AllPay提供給您的特店編號>>";  
/* 取得回傳參數 */  
$arFeedback = $oPayment->CheckOutFeedback();  
/* 檢核與變更訂單狀態 */  
if (sizeof($arFeedback) > 0) {  
    foreach ($arFeedback as $key => $value) {  
        switch ($key)  
        {  
            /* 支付後的回傳的基本參數 */  
            case "MerchantID": $szMerchantID = $value; break;  
            case "MerchantTradeNo": $szMerchantTradeNo = $value; break;  
            case "PaymentDate": $szPaymentDate = $value; break;  
            case "PaymentType": $szPaymentType = $value; break;  
            case "PaymentTypeChargeFee": $szPaymentTypeChargeFee = $value; break;  
            case "RtnCode": $szRtnCode = $value; break;  
            case "RtnMsg": $szRtnMsg = $value; break;  
            case "SimulatePaid": $szSimulatePaid = $value; break;  
            case "TradeAmt": $szTradeAmt = $value; break;  
            case "TradeDate": $szTradeDate = $value; break;  
            case "TradeNo": $szTradeNo = $value; break;  
            default: break;  
        }  
    }  
    // 其他資料處理。  
    .....  
    print '1|OK';  
} else {  
    print '0|Fail';  
}  
}  
catch (Exception $e)  
{  
    // 例外錯誤處理。  
    print '0|' . $e->getMessage();  
}
```

```
}
```

一般付款結果通知範例(JAVA)：

```
/*
 * 接收訂單資料產生完成的範例程式碼。
 */
List<String> enErrors = new ArrayList<String>();
try {
    AllInOne oPayment = new AllInOne();
    oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";
    oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";

    Hashtable<String, String> htFeedback = new Hashtable<String, String>();
    enErrors.addAll(oPayment.CheckOutFeedback(htFeedback, request));

    Set<String> key = htFeedback.keySet();
    String name[] = key.toArray(new String[key.size()]);

    /* 支付後的回傳的基本參數 */
    String szMerchantID = "";
    String szMerchantTradeNo = "";
    String szPaymentDate = "";
    String szPaymentType = "";
    String szPaymentTypeChargeFee = "";
    String szRtnCode = "";
    String szRtnMsg = "";
    String szSimulatePaid = "";
    String szTradeAmt = "";
    String szTradeDate = "";
    String szTradeNo = "";

    // 取得資料
    for(int i = 0 ; i < name.length ; i++) {
        /* 支付後的回傳的基本參數 */
        if(name[i].equals("MerchantID"))
            szMerchantID = htFeedback.get(name[i]);
        else if(name[i].equals("MerchantTradeNo"))
            szMerchantTradeNo = htFeedback.get(name[i]);
    }
}
```

```

        else if(name[i].equals("PaymentDate"))
            szPaymentDate = htFeedback.get(name[i]);
        else if(name[i].equals("PaymentType"))
            szPaymentType = htFeedback.get(name[i]);
        else if(name[i].equals("PaymentTypeChargeFee"))
            szPaymentTypeChargeFee = htFeedback.get(name[i]);
        else if(name[i].equals("RtnCode"))
            szRtnCode = htFeedback.get(name[i]);
        else if(name[i].equals("RtnMsg"))
            szRtnMsg = htFeedback.get(name[i]);
        else if(name[i].equals("SimulatePaid"))
            szSimulatePaid = htFeedback.get(name[i]);
        else if(name[i].equals("TradeAmt"))
            szTradeAmt = htFeedback.get(name[i]);
        else if(name[i].equals("TradeDate"))
            szTradeDate = htFeedback.get(name[i]);
        else if(name[i].equals("TradeNo"))
            szTradeNo = htFeedback.get(name[i]);
    }
    System.out.println("MerchantID = " + szMerchantID);
    System.out.println("MerchantTradeNo = " + szMerchantTradeNo);
    System.out.println("PaymentDate = " + szPaymentDate);
    System.out.println("PaymentType = " + szPaymentType);
    System.out.println("PaymentTypeChargeFee = " + szPaymentTypeChargeFee);
    System.out.println("RtnCode = " + szRtnCode);
    System.out.println("RtnMsg = " + szRtnMsg);
    System.out.println("SimulatePaid = " + szSimulatePaid);
    System.out.println("TradeAmt = " + szTradeAmt);
    System.out.println("TradeDate = " + szTradeDate);
    System.out.println("TradeNo = " + szTradeNo);
    out.println("");
}
catch (Exception e) {
    enErrors.add(e.getMessage());
}
finally {
    // 回覆成功訊息。
    if (enErrors.size() == 0)

```







```

        out.println("I|OK");
    // 回覆錯誤訊息。
    else
        out.println("0" + enErrors);
}
    
```

5.1.2.2. ATM 的取號結果通知參數與範例

使用 ATM 付款方式建立訂單完成後，參數會以 Server POST/Client POST 方式傳送至您設定的 PaymentInfoURL/ClientRedirectURL 位址。

接收 ATM 取號結果通知的參數成員如下表：

參數	名稱	型態	說明
 MerchantID	廠商編號	String(10)	AllPay 提供的廠商編號。
 MerchantTradeNo	廠商交易編號	String(20)	產生訂單時傳送給 AllPay 的廠商交易編號。
 RtnCode	交易狀態	Integer	2 表示取號成功，其餘為失敗(請參考 交易訊息代碼一覽表)。
 RtnMsg	交易訊息	String(200)	
 TradeNo	AllPay 的交易編號	String(20)	請保存 AllPay 的交易編號與 MerchantTradeNo 的關連。
 TradeAmt	交易金額	Decimal	若歐付寶會員選擇信用卡分期付款時，此交易金額會回傳商家在訂單產生時，InstallmentAmount 所帶的金額。
 PaymentType	會員選擇的付款方式	PaymentMethodItem	
 TradeDate	訂單成立時間	DateTime	格式為 yyyy/MM/dd HH:mm:ss。
 BankCode	繳費銀行代碼	String(3)	
 vAccount	繳費虛擬帳號	String(16)	
 ExpireDate	繳費期限	String(10)	格式為 yyyy/MM/dd。

ATM 取號結果通知範例(C#)：

```

/*
 * 接收訂單資料產生完成的範例程式碼。
 */
List<string> enErrors = new List<string>();
Hashtable htFeedback = null;
    
```

```
try
{
    using (AllInOne oPayment = new AllInOne())
    {
        oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";
        oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";
        /* 取回付款結果 */
        enErrors.AddRange(oPayment.CheckOutFeedback(ref htFeedback));
    }
    // 取回所有資料
    if (enErrors.Count() == 0)
    {
        /* 支付後的回傳的基本參數 */
        string szMerchantID = String.Empty;
        string szMerchantTradeNo = String.Empty;
        string szPaymentType = String.Empty;
        string szRtnCode = String.Empty;
        string szRtnMsg = String.Empty;
        string szTradeAmt = String.Empty;
        string szTradeDate = String.Empty;
        string szTradeNo = String.Empty;
        /* 使用 ATM 交易時，回傳的額外參數 */
        string szBankCode = String.Empty;
        string szVirtualAccount = String.Empty;
        string szExpireDate = String.Empty;
        // 取得資料於畫面
        foreach (string szKey in htFeedback.Keys)
        {
            switch (szKey)
            {
                /* 使用 ATM 交易時回傳的參數 */
                case "MerchantID": szMerchantID = this.Request.Form[szKey]; break;
                case "MerchantTradeNo": szMerchantTradeNo = this.Request.Form[szKey]; break;
                case "RtnCode": szRtnCode = this.Request.Form[szKey]; break;
                case "RtnMsg": szRtnMsg = this.Request.Form[szKey]; break;
                case "TradeNo": szTradeNo = this.Request.Form[szKey]; break;
                case "TradeAmt": szTradeAmt = this.Request.Form[szKey]; break;
```

```

        case "PaymentType": szPaymentType = this.Request.Form[szKey]; break;
        case "TradeDate": szTradeDate = this.Request.Form[szKey]; break;
        case "BankCode": szBankCode = this.Request.Form[szKey]; break;
        case "vAccount": szVirtualAccount = this.Request.Form[szKey]; break;
        case "ExpireDate": szExpireDate = this.Request.Form[szKey]; break;
        default: break;
    }
}
// 其他資料處理。
.....
} else {
    // 其他資料處理。
    .....
}
}
catch (Exception ex)
{
    // 例外錯誤處理。
    enErrors.Add(ex.Message);
}
finally
{
    this.Response.Clear();
    // 回覆成功訊息。
    if (enErrors.Count() == 0)
        this.Response.Write("1|OK");
    // 回覆錯誤訊息。
    else
        this.Response.Write(String.Format("0|{0}", String.Join("\r\n", enErrors)));
    this.Response.Flush();
    this.Response.End();
}

```

ATM 取號結果通知範例(PHP)：

```

/*
 * 接收訂單資料產生完成的範例程式碼。
 */
try

```

```
{  
    $oPayment = new AllInOne();  
    /* 服務參數 */  
    $oPayment->HashKey = "<<AllPay提供給您的Hash Key>>";  
    $oPayment->HashIV = "<<AllPay提供給您的Hash IV>>";  
    /* 取得回傳參數 */  
    $arFeedback = $oPayment->CheckOutFeedback();  
    /* 檢核與變更訂單狀態 */  
    if (sizeof($arFeedback) > 0) {  
        foreach ($arFeedback as $key => $value) {  
            switch ($key)  
            {  
                /* 使用 ATM 交易時回傳的參數 */  
                case "MerchantID": $szMerchantID = $value; break;  
                case "MerchantTradeNo": $szMerchantTradeNo = $value; break;  
                case "RtnCode": $szRtnCode = $value; break;  
                case "RtnMsg": $szRtnMsg = $value; break;  
                case "TradeNo": $szTradeNo = $value; break;  
                case "TradeAmt": $szTradeAmt = $value; break;  
                case "PaymentType": $szPaymentType = $value; break;  
                case "TradeDate": $szTradeDate = $value; break;  
                case "BankCode": $szBankCode = $value; break;  
                case "vAccount": $szVirtualAccount = $value; break;  
                case "ExpireDate": $szExpireDate = $value; break;  
                default: break;  
            }  
        }  
        // 其他資料處理。  
        .....  
        print '1|OK';  
    } else {  
        print '0|Fail';  
    }  
}  
catch (Exception $e)  
{  
    // 例外錯誤處理。  
    print '0|' . $e->getMessage();  
}
```

```
}
```

ATM 取號結果通知範例(JAVA) :

```
/*
 * 接收訂單資料產生完成的範例程式碼。
 */
List<String> enErrors = new ArrayList<String>();
try {
    AllInOne oPayment = new AllInOne();
    oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";
    oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";

    Hashtable<String, String> htFeedback = new Hashtable<String, String>();
    enErrors.addAll(oPayment.CheckOutFeedback(htFeedback, request));

    Set<String> key = htFeedback.keySet();
    String name[] = key.toArray(new String[key.size()]);

    /* 支付後的回傳的基本參數 */
    String szMerchantID = "";
    String szMerchantTradeNo = "";
    String szPaymentDate = "";
    String szPaymentType = "";
    String szPaymentTypeChargeFee = "";
    String szRtnCode = "";
    String szRtnMsg = "";
    String szSimulatePaid = "";
    String szTradeAmt = "";
    String szTradeDate = "";
    String szTradeNo = "";
    /* 使用 ATM 交易時 · 回傳的額外參數 */
    String szBankCode = "";
    String szVirtualAccount = "";
    String szExpireDate = "";

    // 取得資料
    for(int i = 0 ; i < name.length ; i++) {
        /* 支付後的回傳的基本參數 */

```

```

        if(name[i].equals("MerchantID"))
            szMerchantID = htFeedback.get(name[i]);
        else if(name[i].equals("MerchantTradeNo"))
            szMerchantTradeNo = htFeedback.get(name[i]);
        else if(name[i].equals("PaymentDate"))
            szPaymentDate = htFeedback.get(name[i]);
        else if(name[i].equals("PaymentType"))
            szPaymentType = htFeedback.get(name[i]);
        else if(name[i].equals("PaymentTypeChargeFee"))
            szPaymentTypeChargeFee = htFeedback.get(name[i]);
        else if(name[i].equals("RtnCode"))
            szRtnCode = htFeedback.get(name[i]);
        else if(name[i].equals("RtnMsg"))
            szRtnMsg = htFeedback.get(name[i]);
        else if(name[i].equals("SimulatePaid"))
            szSimulatePaid = htFeedback.get(name[i]);
        else if(name[i].equals("TradeAmt"))
            szTradeAmt = htFeedback.get(name[i]);
        else if(name[i].equals("TradeDate"))
            szTradeDate = htFeedback.get(name[i]);
        else if(name[i].equals("TradeNo"))
            szTradeNo = htFeedback.get(name[i]);
        else if(name[i].equals("BankCode"))
            szBankCode = htFeedback.get(name[i]);
        else if(name[i].equals("vAccount"))
            szVirtualAccount = htFeedback.get(name[i]);
        else if(name[i].equals("ExpireDate"))
            szExpireDate = htFeedback.get(name[i]);
    }

    System.out.println("MerchantID = " + szMerchantID);
    System.out.println("MerchantTradeNo = " + szMerchantTradeNo);
    System.out.println("PaymentDate = " + szPaymentDate);
    System.out.println("PaymentType = " + szPaymentType);
    System.out.println("PaymentTypeChargeFee = " + szPaymentTypeChargeFee);
    System.out.println("RtnCode = " + szRtnCode);
    System.out.println("RtnMsg = " + szRtnMsg);
    System.out.println("SimulatePaid = " + szSimulatePaid);
    System.out.println("TradeAmt = " + szTradeAmt);
    
```

```

        System.out.println("TradeDate = " + szTradeDate);
        System.out.println("TradeNo = " + szTradeNo);
        /* 使用 ATM 交易時 · 回傳的額外參數 */
        System.out.println("BankCode = " + szBankCode);
        System.out.println("vAccount = " + szVirtualAccount);
        System.out.println("ExpireDate = " + szExpireDate);
        out.println("");
    }
    catch (Exception e) {
        enErrors.add(e.getMessage());
    }
    finally {
        // 回覆成功訊息。
        if (enErrors.size() == 0)
            out.println("1|OK");
        // 回覆錯誤訊息。
        else
            out.println("0|" + enErrors);
    }
}

```

5.1.2.3. CVS 或 BARCODE 的取號結果通知參數與範例

使用 CVS 或 BARCODE 付款方式建立訂單完成後，參數會以 Server POST/Client POST 方式傳送至您設定的 PaymentInfoURL/CleintRedirectURL 位址。

接收 CVS 或 BARCODE 取號結果通知的參數成員如下表：

	參數	名稱	型態	說明
	MerchantID	廠商編號	String(10)	AllPay 提供的廠商編號。
	MerchantTradeNo	廠商交易編號	String(20)	產生訂單時傳送給 AllPay 的廠商交易編號。
	RtnCode	交易狀態	Integer	10100073 表示取號成功，其餘代碼為失敗(請參考 交易訊息代碼一覽表)。
	RtnMsg	交易訊息	String(200)	
	TradeNo	AllPay 的交易編號	String(20)	請保存 AllPay 的交易編號與 MerchantTradeNo 的關連。
	TradeAmt	交易金額	Decimal	若歐付寶會員選擇信用卡分期付款時，此交易金額會回傳商家在

				訂單產生時， InstallmentAmount 所帶的金額。
	PaymentType	會員選擇的付款方式	PaymentMethodItem	
	TradeDate	訂單成立時間	DateTime	格式為 yyyy/MM/dd HH:mm:ss。
	PaymentNo	繳費代碼	String(14)	如果是條碼，則此欄位回傳空白。
	ExpireDate	繳費期限	DateTime	格式為 yyyy/MM/dd HH:mm:ss
	Barcode1	條碼第一段號碼	String(20)	格式為 9 碼數字，如果是代碼，則此欄位回傳空白。
	Barcode2	條碼第二段號碼	String(20)	格式為 16 碼數字，如果是代碼，則此欄位回傳空白。
	Barcode3	條碼第三段號碼	String(20)	格式為 15 碼數字，如果是代碼，則此欄位回傳空白。

CVS 或 BARCODE 取號結果通知範例(C#)：

```

/*
 * 接收訂單資料產生完成的範例程式碼。
 */
List<string> enErrors = new List<string>();
Hashtable htFeedback = null;

try
{
    using (AllInOne oPayment = new AllInOne())
    {
        oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";
        oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";
        /* 取回付款結果 */
        enErrors.AddRange(oPayment.CheckOutFeedback(ref htFeedback));
    }
    // 取回所有資料
    if (enErrors.Count() == 0)
    {
        /* 支付後的回傳的基本參數 */
        string szMerchantID = String.Empty;
        string szMerchantTradeNo = String.Empty;
        string szPaymentType = String.Empty;
        string szRtnCode = String.Empty;
    }
}

```



```
string szRtnMsg = String.Empty;
string szTradeAmt = String.Empty;
string szTradeDate = String.Empty;
string szTradeNo = String.Empty;
/* 使用 CVS 或 BARCODE 交易時，回傳的額外參數 */
string szPaymentNo = String.Empty;
string szExpireDate = String.Empty;
string szBarcode1 = String.Empty;
string szBarcode2 = String.Empty;
string szBarcode3 = String.Empty;
// 取得資料於畫面
foreach (string szKey in htFeedback.Keys)
{
    switch (szKey)
    {
        /* 使用 CVS/BARCODE 交易時回傳的參數 */
        case "MerchantID": szMerchantID = this.Request.Form[szKey]; break;
        case "MerchantTradeNo": szMerchantTradeNo = this.Request.Form[szKey]; break;
        case "RtnCode": szRtnCode = this.Request.Form[szKey]; break;
        case "RtnMsg": szRtnMsg = this.Request.Form[szKey]; break;
        case "TradeNo": szTradeNo = this.Request.Form[szKey]; break;
        case "TradeAmt": szTradeAmt = this.Request.Form[szKey]; break;
        case "PaymentType": szPaymentType = this.Request.Form[szKey]; break;
        case "TradeDate": szTradeDate = this.Request.Form[szKey]; break;
        case "PaymentNo": szPaymentNo = this.Request.Form[szKey]; break;
        case "ExpireDate": szExpireDate = this.Request.Form[szKey]; break;
        case "Barcode1": szBarcode1 = this.Request.Form[szKey]; break;
        case "Barcode2": szBarcode2 = this.Request.Form[szKey]; break;
        case "Barcode3": szBarcode3 = this.Request.Form[szKey]; break;
        default: break;
    }
}
// 其他資料處理。
.....
} else {
    // 其他資料處理。
    .....
}
```

```

}
catch (Exception ex)
{
    // 例外錯誤處理。
    enErrors.Add(ex.Message);
}
finally
{
    this.Response.Clear();
    // 回覆成功訊息。
    if (enErrors.Count() == 0)
        this.Response.Write("I|OK");
    // 回覆錯誤訊息。
    else
        this.Response.Write(String.Format("0|{0}", String.Join("\r\n", enErrors)));
    this.Response.Flush();
    this.Response.End();
}

```

CVS 或 BARCODE 取號結果通知範例(PHP) :

```

/*
 * 接收訂單資料產生完成的範例程式碼。
 */
try
{
    $oPayment = new AllInOne();
    /* 服務參數 */
    $oPayment->HashKey = "<<AllPay提供給您的Hash Key>>";
    $oPayment->HashIV = "<<AllPay提供給您的Hash IV>>";
    /* 取得回傳參數 */
    $arFeedback = $oPayment->CheckOutFeedback();
    /* 檢核與變更訂單狀態 */
    if (sizeof($arFeedback) > 0) {
        foreach ($arFeedback as $key => $value) {
            switch ($key)
            {
                /* 使用 CVS/BARCODE 交易時回傳的參數 */
                case "MerchantID": $szMerchantID = $value; break;
            }
        }
    }
}

```

```

        case "MerchantTradeNo": $szMerchantTradeNo = $value; break;
        case "RtnCode": $szRtnCode = $value; break;
        case "RtnMsg": $szRtnMsg = $value; break;
        case "TradeNo": $szTradeNo = $value; break;
        case "TradeAmt": $szTradeAmt = $value; break;
        case "PaymentType": $szPaymentType = $value; break;
        case "TradeDate": $szTradeDate = $value; break;
        case "PaymentNo": $szPaymentNo = $value; break;
        case "Barcode1": $szBarcode1 = $value; break;
        case "Barcode2": $szBarcode2 = $value; break;
        case "Barcode3": $szBarcode3 = $value; break;
        default: break;
    }
}
// 其他資料處理。
.....
print '1|OK';
} else {
    print '0|Fail';
}
}
catch (Exception $e)
{
    // 例外錯誤處理。
    print '0|' . $e->getMessage();
}

```

CVS 或 BARCODE 取號結果通知範例(JAVA)：

```

/*
 * 接收訂單資料產生完成的範例程式碼。
 */
List<String> enErrors = new ArrayList<String>();
try {
    AllInOne oPayment = new AllInOne();
    oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";
    oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";

    Hashtable<String, String> htFeedback = new Hashtable<String, String>();

```

```
enErrors.addAll(oPayment.CheckOutFeedback(htFeedback, request));
```

```
Set<String> key = htFeedback.keySet();
```

```
String name[] = key.toArray(new String[key.size()]);
```

```
/* 支付後的回傳的基本參數 */
```

```
String szMerchantID = "";
```

```
String szMerchantTradeNo = "";
```

```
String szPaymentDate = "";
```

```
String szPaymentType = "";
```

```
String szPaymentTypeChargeFee = "";
```

```
String szRtnCode = "";
```

```
String szRtnMsg = "";
```

```
String szSimulatePaid = "";
```

```
String szTradeAmt = "";
```

```
String szTradeDate = "";
```

```
String szTradeNo = "";
```

```
/* 使用 CVS 或 BARCODE 交易時，回傳的額外參數 */
```

```
String szPaymentNo = "";
```

```
String szExpireDate = "";
```

```
String szBarcode1 = "";
```

```
String szBarcode2 = "";
```

```
String szBarcode3 = "";
```

```
// 取得資料
```

```
for(int i = 0 ; i < name.length ; i++) {
```

```
    /* 支付後的回傳的基本參數 */
```

```
    if(name[i].equals("MerchantID"))
```

```
        szMerchantID = htFeedback.get(name[i]);
```

```
    else if(name[i].equals("MerchantTradeNo"))
```

```
        szMerchantTradeNo = htFeedback.get(name[i]);
```

```
    else if(name[i].equals("PaymentDate"))
```

```
        szPaymentDate = htFeedback.get(name[i]);
```

```
    else if(name[i].equals("PaymentType"))
```

```
        szPaymentType = htFeedback.get(name[i]);
```

```
    else if(name[i].equals("PaymentTypeChargeFee"))
```

```
        szPaymentTypeChargeFee = htFeedback.get(name[i]);
```

```
    else if(name[i].equals("RtnCode"))
```

```
        szRtnCode = htFeedback.get(name[i]);
    else if(name[i].equals("RtnMsg"))
        szRtnMsg = htFeedback.get(name[i]);
    else if(name[i].equals("SimulatePaid"))
        szSimulatePaid = htFeedback.get(name[i]);
    else if(name[i].equals("TradeAmt"))
        szTradeAmt = htFeedback.get(name[i]);
    else if(name[i].equals("TradeDate"))
        szTradeDate = htFeedback.get(name[i]);
    else if(name[i].equals("TradeNo"))
        szTradeNo = htFeedback.get(name[i]);
    else if(name[i].equals("PaymentNo"))
        szPaymentNo = htFeedback.get(name[i]);
    else if(name[i].equals("ExpireDate"))
        szExpireDate = htFeedback.get(name[i]);
    else if(name[i].equals("Barcode1"))
        szBarcode1 = htFeedback.get(name[i]);
    else if(name[i].equals("Barcode2"))
        szBarcode2 = htFeedback.get(name[i]);
    else if(name[i].equals("Barcode3"))
        szBarcode3 = htFeedback.get(name[i]);
}

System.out.println("MerchantID = " + szMerchantID);
System.out.println("MerchantTradeNo = " + szMerchantTradeNo);
System.out.println("PaymentDate = " + szPaymentDate);
System.out.println("PaymentType = " + szPaymentType);
System.out.println("PaymentTypeChargeFee = " + szPaymentTypeChargeFee);
System.out.println("RtnCode = " + szRtnCode);
System.out.println("RtnMsg = " + szRtnMsg);
System.out.println("SimulatePaid = " + szSimulatePaid);
System.out.println("TradeAmt = " + szTradeAmt);
System.out.println("TradeDate = " + szTradeDate);
System.out.println("TradeNo = " + szTradeNo);
/* 使用 CVS 或 BARCODE 交易時，回傳的額外參數 */
System.out.println("PaymentNo = " + szPaymentNo);
System.out.println("ExpireDate = " + szExpireDate);
System.out.println("Barcode1 = " + szBarcode1);
System.out.println("Barcode2 = " + szBarcode2);
```

```

        System.out.println("Barcode3 = " + szBarcode3);
        out.println("");
    }
    catch (Exception e) {
        enErrors.add(e.getMessage());
    }
    finally {
        // 回覆成功訊息。
        if (enErrors.size() == 0)
            out.println("I|OK");
        // 回覆錯誤訊息。
        else
            out.println("0" + enErrors);
    }
}


```

5.1.2.4. Credit 定期定額的授權成功通知參數與範例

當訂單是使用信用卡定期定額的交易時，在每次授權成功後，以下參數會以 Server POST 方式傳送至您設定的 PeriodReturnURL 位址。

接收 Credit 定期定額授權成功結果通知的參數成員如下表：

參數	名稱	型態	說明
 MerchantID	廠商編號	String(10)	
 MerchantTradeNo	廠商交易編號	String(20)	訂單產生時傳送給 AllPay 的廠商交易編號
 RtnCode	交易狀態	Integer	1 表示授權成功，其餘為失敗(請參考 交易訊息代碼一覽表)。
 RtnMsg	交易訊息	String(200)	交易訊息
 PeriodType	週期種類	PeriodType	訂單建立時的所設定的週期種類
 Frequency	執行頻率	Integer	訂單建立時的所設定的執行頻率
 ExecTimes	執行次數	Integer	訂單建立時的所設定的執行頻率
 Amount	本次授權金額	Integer	此次所授權的金額
 Gwsr	授權交易單號	Integer	此次所授權的交易單號
 ProcessDate	處理時間	String(20)	處理時間 (yyyy/MM/dd HH:mm:ss)
 AuthCode	授權碼	String(6)	授權碼
 FirstAuthAmount	初次授權金額	Decimal	第一次授權的金額。 因為要是以交易金額(TotalAmount)來做

				定期定額，分期後差額部分於第一期收取。
	TotalSuccessTimes	已執行成功次數	Integer	目前已成功授權的次數。

Credit 定期定額授權成功結果通知範例(C#)：

```

/*
 * 接收訂單資料產生完成的範例程式碼。
 */
List<string> enErrors = new List<string>();
Hashtable htFeedback = null;

try
{
    using (AllInOne oPayment = new AllInOne())
    {
        oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";
        oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";
        /* 取回付款結果 */
        enErrors.AddRange(oPayment.CheckOutFeedback(ref htFeedback));
    }
    // 取回所有資料
    if (enErrors.Count() == 0)
    {
        /* 支付後的回傳的基本參數 */
        string szMerchantID = String.Empty;
        string szMerchantTradeNo = String.Empty;
        string szRtnCode = String.Empty;
        string szRtnMsg = String.Empty;
        /* 使用定期定額交易時，回傳的額外參數 */
        string szPeriodType = String.Empty;
        string szFrequency = String.Empty;
        string szExecTimes = String.Empty;
        string szAmount = String.Empty;
        string szGwsr = String.Empty;
        string szProcessDate = String.Empty;
        string szAuthCode = String.Empty;
        string szFirstAuthAmount = String.Empty;
        string szTotalSuccessTimes = String.Empty;
    }
}

```

```
// 取得資料於畫面
foreach (string szKey in htFeedback.Keys)
{
    switch (szKey)
    {
        /* 使用定期定額交易時回傳的參數 */
        case "MerchantID": szMerchantID = this.Request.Form[szKey]; break;
        case "MerchantTradeNo": szMerchantTradeNo = this.Request.Form[szKey]; break;
        case "RtnCode": szRtnCode = this.Request.Form[szKey]; break;
        case "RtnMsg": szRtnMsg = this.Request.Form[szKey]; break;
        case "PeriodType": szPeriodType = this.Request.Form[szKey]; break;
        case "Frequency": szFrequency = this.Request.Form[szKey]; break;
        case "ExecTimes": szExecTimes = this.Request.Form[szKey]; break;
        case "Amount": szAmount = this.Request.Form[szKey]; break;
        case "Gwsr": szGwsr = this.Request.Form[szKey]; break;
        case "ProcessDate": szProcessDate = this.Request.Form[szKey]; break;
        case "AuthCode": szAuthCode = this.Request.Form[szKey]; break;
        case "FirstAuthAmount": szFirstAuthAmount = this.Request.Form[szKey]; break;
        case "TotalSuccessTimes": szTotalSuccessTimes = this.Request.Form[szKey]; break;
        default: break;
    }
}

// 其他資料處理。
.....

} else {
    // 其他資料處理。
    .....
}

}

catch (Exception ex)
{
    enErrors.Add(ex.Message);
}

finally
{
    this.Response.Clear();
    // 回覆成功訊息。
    if (enErrors.Count() == 0)
```



```
        this.Response.Write("I|OK");  
    // 回覆錯誤訊息。  
    else  
        this.Response.Write(String.Format("O|{0}", String.Join("\\r\\n", enErrors)));  
    this.Response.Flush();  
    this.Response.End();  
}
```

Credit 定期定額授權成功結果通知範例(PHP)：

```
/*  
 * 接收訂單資料產生完成的範例程式碼。  
 */  
try  
{  
    $oPayment = new AllInOne();  
    /* 服務參數 */  
    $oPayment->HashKey = "<<AllPay提供給您的Hash Key>>";  
    $oPayment->HashIV = "<<AllPay提供給您的Hash IV>>";  
    /* 取得回傳參數 */  
    $arFeedback = $oPayment->CheckOutFeedback();  
    /* 檢核與變更訂單狀態 */  
    if (sizeof($arFeedback) > 0) {  
        foreach ($arFeedback as $key => $value) {  
            switch ($key)  
            {  
                /* 使用 Credit 定期定額交易時回傳的參數 */  
                case "MerchantID": $szMerchantID = $value; break;  
                case "MerchantTradeNo": $szMerchantTradeNo = $value; break;  
                case "RtnCode": $szRtnCode = $value; break;  
                case "RtnMsg": $szRtnMsg = $value; break;  
                case "PeriodType": $szPeriodType = $value; break;  
                case "Frequency": $szFrequency = $value; break;  
                case "ExecTimes": $szExecTimes = $value; break;  
                case "Amount": $szAmount = $value; break;  
                case "Gwsr": $szGwsr = $value; break;  
                case "ProcessDate": $szProcessDate = $value; break;  
                case "AuthCode": $szAuthCode = $value; break;  
                case "FirstAuthAmount": $szFirstAuthAmount = $value; break;  
            }  
        }  
    }  
}
```

```

        case "TotalSuccessTimes": $szTotalSuccessTimes = $value; break;
        default: break;
    }
}
// 其他資料處理。
.....
print '1|OK';
} else {
    print '0|Fail';
}
}
catch (Exception $e)
{
    // 例外錯誤處理。
    print '0|' . $e->getMessage();
}

```

Credit 定期定額授權成功結果通知範例(JAVA)：

```

/*
 * 接收訂單資料產生完成的範例程式碼。
 */
List<String> enErrors = new ArrayList<String>();
try {
    AllInOne oPayment = new AllInOne();
    oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";
    oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";

    Hashtable<String, String> htFeedback = new Hashtable<String, String>();
    enErrors.addAll(oPayment.CheckOutFeedback(htFeedback, request));

    Set<String> key = htFeedback.keySet();
    String name[] = key.toArray(new String[key.size()]);

    /* 支付後的回傳的基本參數 */
    String szMerchantID = "";
    String szMerchantTradeNo = "";
    String szPaymentDate = "";
    String szPaymentType = "";
}

```

```
String szPaymentTypeChargeFee = "";
String szRtnCode = "";
String szRtnMsg = "";
String szSimulatePaid = "";
String szTradeAmt = "";
String szTradeDate = "";
String szTradeNo = "";
/* 使用定期定額交易時，回傳的額外參數 */
String szPeriodType = "";
String szFrequency = "";
String szExecTimes = "";
String szAmount = "";
String szGwsr = "";
String szProcessDate = "";
String szAuthCode = "";
String szFirstAuthAmount = "";
String szTotalSuccessTimes = "";
// 取得資料
for(int i = 0 ; i < name.length ; i++) {
    /* 支付後的回傳的基本參數 */
    if(name[i].equals("MerchantID"))
        szMerchantID = htFeedback.get(name[i]);
    else if(name[i].equals("MerchantTradeNo"))
        szMerchantTradeNo = htFeedback.get(name[i]);
    else if(name[i].equals("PaymentDate"))
        szPaymentDate = htFeedback.get(name[i]);
    else if(name[i].equals("PaymentType"))
        szPaymentType = htFeedback.get(name[i]);
    else if(name[i].equals("PaymentTypeChargeFee"))
        szPaymentTypeChargeFee = htFeedback.get(name[i]);
    else if(name[i].equals("RtnCode"))
        szRtnCode = htFeedback.get(name[i]);
    else if(name[i].equals("RtnMsg"))
        szRtnMsg = htFeedback.get(name[i]);
    else if(name[i].equals("SimulatePaid"))
        szSimulatePaid = htFeedback.get(name[i]);
    else if(name[i].equals("TradeAmt"))
        szTradeAmt = htFeedback.get(name[i]);
}
```

```

        else if(name[i].equals("TradeDate"))
            szTradeDate = htFeedback.get(name[i]);
        else if(name[i].equals("TradeNo"))
            szTradeNo = htFeedback.get(name[i]);
        else if(name[i].equals("PeriodType"))
            szPeriodType = htFeedback.get(name[i]);
        else if(name[i].equals("Frequency"))
            szFrequency = htFeedback.get(name[i]);
        else if(name[i].equals("ExecTimes"))
            szExecTimes = htFeedback.get(name[i]);
        else if(name[i].equals("Amount"))
            szAmount = htFeedback.get(name[i]);
        else if(name[i].equals("Gwsr"))
            szGwsr = htFeedback.get(name[i]);
        else if(name[i].equals("ProcessDate"))
            szProcessDate = htFeedback.get(name[i]);
        else if(name[i].equals("AuthCode"))
            szAuthCode = htFeedback.get(name[i]);
        else if(name[i].equals("FirstAuthAmount"))
            szFirstAuthAmount = htFeedback.get(name[i]);
        else if(name[i].equals("TotalSuccessTimes"))
            szTotalSuccessTimes = htFeedback.get(name[i]);
    }

    System.out.println("MerchantID = " + szMerchantID);
    System.out.println("MerchantTradeNo = " + szMerchantTradeNo);
    System.out.println("PaymentDate = " + szPaymentDate);
    System.out.println("PaymentType = " + szPaymentType);
    System.out.println("PaymentTypeChargeFee = " + szPaymentTypeChargeFee);
    System.out.println("RtnCode = " + szRtnCode);
    System.out.println("RtnMsg = " + szRtnMsg);
    System.out.println("SimulatePaid = " + szSimulatePaid);
    System.out.println("TradeAmt = " + szTradeAmt);
    System.out.println("TradeDate = " + szTradeDate);
    System.out.println("TradeNo = " + szTradeNo);
    /* 使用定期定額交易時，回傳的額外參數 */
    System.out.println("PeriodType = " + szPeriodType);
    System.out.println("Frequency = " + szFrequency);
    System.out.println("ExecTimes = " + szExecTimes);

```

```

        System.out.println("Amount = " + szAmount);
        System.out.println("Gwsr = " + szGwsr);
        System.out.println("ProcessDate = " + szProcessDate);
        System.out.println("AuthCode = " + szAuthCode);
        System.out.println("FirstAuthAmount = " + szFirstAuthAmount);
        System.out.println("TotalSuccessTimes = " + szTotalSuccessTimes);
        out.println("");
    }
    catch (Exception e) {
        enErrors.add(e.getMessage());
    }
    finally {
        // 回覆成功訊息。
        if (enErrors.size() == 0)
            out.println("1|OK");
        // 回覆錯誤訊息。
        else
            out.println("0|" + enErrors);
    }
}

```

5.1.3. 訂單查詢的方法(QueryTradeInfo)

本方法提供特店(賣家會員)進行訂單查詢的作業；在介接該服務之前，請至「廠商後台 > 基本資料查詢 > 廠商基本資料」設定加入特店伺服器的對外 IP 位址到「允許的 IP」欄位([請參考](#))。

介接的服務網址(正式機為 https，測試機為 http)：

Http POST

- 正式環境：<https://payment.allpay.com.tw/Cashier/QueryTradeInfo>
- 測試環境：<http://payment-stage.allpay.com.tw/Cashier/QueryTradeInfo>

Http SOAP(Web Service) *不支援 PHP 版本

- 正式環境：<https://payment.allpay.com.tw/api/purchase/allpay.asmx?op=QueryTradeInfo>
- 測試環境：<http://payment-stage.allpay.com.tw/api/purchase/allpay.asmx?op=QueryTradeInfo>

5.1.3.1. 訂單查詢使用參數

基本資料傳送的參數成員如下表(名稱前若有紅色星號*為必填欄位)：

	名稱	型態	說明
	*Query.MerchantTradeNo	String(20)	廠商的交易編號。


接收基本回傳的參數成員如下表：

	參數	名稱	型態	說明
	MerchantID	廠商編號	String(10)	AllPay 提供的廠商編號。
	MerchantTradeNo	廠商交易編號	String(20)	產生訂單時傳送給 AllPay 的廠商交易編號。
	TradeNo	AllPay 的交易編號	String(20)	請保存 AllPay 的交易編號與 MerchantTradeNo 的關連。
	TradeAmt	交易金額	Decimal	若歐付寶會員選擇信用卡分期付款時，此交易金額會回傳商家在訂單產生時，InstallmentAmount 所帶的金額。
	PaymentDate	付款時間	DateTime	格式為 yyyy/MM/dd HH:mm:ss。
	PaymentType	會員選擇的付款方式	PaymentMethodItem	
	HandlingCharge	手續費合計	Decimal	
	PaymentType ChargeFee	通路費	Decimal	
	TradeDate	訂單成立時間	DateTime	格式為 yyyy/MM/dd HH:mm:ss。
	TradeStatus	交易狀態	Integer	請參考 交易狀態代碼一覽表 。
	ItemName	商品名稱	String(200)	

接收延伸回傳的參數成員如下表：

當付款方式設定為 WebATM 時				
	參數	名稱	型態	說明
	WebATMAccBank	付款人銀行代碼	String(3)	
	WebATMAccNo	付款人銀行帳號後五碼	String(5)	
	WebATMBankName	銀行名稱	String(10)	
當付款方式設定為 ATM 時				
	參數	名稱	型態	說明
	ATMAccBank	付款人銀行代碼	String(3)	
	ATMAccNo	付款人銀行帳號後五碼	String(5)	
當付款方式設定為 CVS 或 BARCODE 時				

	參數	名稱	型態	說明
	PaymentNo	繳費代碼	String(14)	
	PayFrom	繳費超商	String(10)	
當付款方式設定為 Alipay 時				
	參數	名稱	型態	說明
	AlipayID	付款人在支付寶的系統編號	String(10)	
	AlipayTradeNo	支付寶交易編號	String(10)	
當付款方式設定為 Tenpay 時				
	參數	名稱	型態	說明
	TenpayTradeNo	財付通交易編號	String(20)	
當付款方式設定為 Credit 時				
	參數	名稱	型態	說明
	gwsr	授權交易單號	Integer	
	process_date	處理時間	DateTime	格式為 yyyy/MM/dd HH:mm:ss。
	auth_code	授權碼	String(6)	
	amount	金額	Integer	
	stage	分期期數	Integer	
	stast	頭期金額	Integer	
	staed	各期金額	Integer	
	eci	3D(VBV)	Integer	回傳值(eci=5,6,2,1 代表該筆交易不可否認)
	card4no	卡片的末 4 碼	String(4)	
	card6no	卡片的前 6 碼	String(6)	
	red_dan	紅利扣點	Integer	
	red_de_amt	紅利折抵金額	Integer	
	red_ok_amt	實際扣款金額	Integer	
	red_yet	紅利剩餘點數	Integer	
	PeriodType	訂單建立時的所設定的週期種類	PeriodType	
	Frequency	訂單建立時的所設定的執行頻率	Integer	
	ExecTimes	訂單建立時的所設定的執行頻率	Integer	
	PeriodAmount	訂單建立時的每次要授權金額	Integer	
	TotalSuccessTimes	目前已成功授權的次數	Integer	

	TotalSuccessAmount	目前已成功授權的金額 合計	Integer	
---	--------------------	------------------	---------	--

5.1.3.2. 訂單查詢程式碼範例

訂單查詢程式碼範例(C#)：

```

/*
 * 查詢訂單的範例程式碼。
 */
List<string> enErrors = new List<string>();
Hashtable htFeedback = null;

try
{
    using (AllInOne oPayment = new AllInOne())
    {
        /* 服務參數 */
        oPayment.ServiceMethod = HttpMethod.ServerPOST; // 或使用 HttpMethod.HttpSOAP;
        oPayment.ServiceURL = "<<您要呼叫的服務位址>>";
        oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";
        oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";
        oPayment.MerchantID = "<<AllPay提供給您的特店編號>>";
        /* 基本參數 */
        oPayment.Query.MerchantTradeNo = "<<您要查詢的訂單交易編號>>";
        /* 查詢訂單 */
        enErrors.AddRange(oPayment.QueryTradeInfo(ref htFeedback));
    }
    // 取回所有資料
    if (enErrors.Count() == 0)
    {
        /* 查詢後的回傳的基本參數 */
        string szMerchantID = String.Empty;
        string szMerchantTradeNo = String.Empty;
        string szTradeNo = String.Empty;
        string szTradeAmt = String.Empty;
        string szPaymentDate = String.Empty;
        string szPaymentType = String.Empty;
        string szHandlingCharge = String.Empty;
    }
}

```



```
string szPaymentTypeChargeFee = String.Empty;
string szTradeDate = String.Empty;
string szTradeStatus = String.Empty;
string szItemName = String.Empty;
/* 使用 WebATM 交易時，回傳的額外參數 */
string szWebATMAccBank = String.Empty;
string szWebATMAccNo = String.Empty;
/* 使用 ATM 交易時，回傳的額外參數 */
string szATMAccBank = String.Empty;
string szATMAccNo = String.Empty;
/* 使用 CVS 或 BARCODE 交易時，回傳的額外參數 */
string szPaymentNo = String.Empty;
string szPayFrom = String.Empty;
/* 使用 Alipay 交易時，回傳的額外參數 */
string szAlipayID = String.Empty;
string szAlipayTradeNo = String.Empty;
/* 使用 Tenpay 交易時，回傳的額外參數 */
string szTenpayTradeNo = String.Empty;
/* 使用 Credit 交易時，回傳的額外參數 */
string szGwsr = String.Empty;
string szProcessDate = String.Empty;
string szAuthCode = String.Empty;
string szAmount = String.Empty;
string szStage = String.Empty;
string szStast = String.Empty;
string szStaed = String.Empty;
string szECI = String.Empty;
string szCard4No = String.Empty;
string szCard6No = String.Empty;
string szRedDan = String.Empty;
string szRedDeAmt = String.Empty;
string szRedOkAmt = String.Empty;
string szRedYet = String.Empty;
string szPeriodType = String.Empty;
string szFrequency = String.Empty;
string szExecTimes = String.Empty;
string szPeriodAmount = String.Empty;
string szTotalSuccessTimes = String.Empty;
```

```
string szTotalSuccessAmount = String.Empty;
// 取得資料於畫面
foreach (string szKey in htFeedback.Keys)
{
    switch (szKey)
    {
        /* 查詢後的回傳的基本參數 */
        case "MerchantID": szMerchantID = this.Request.Form[szKey]; break;
        case "MerchantTradeNo": szMerchantTradeNo = this.Request.Form[szKey]; break;
        case "TradeNo": szTradeNo = this.Request.Form[szKey]; break;
        case "TradeAmt": szTradeAmt = this.Request.Form[szKey]; break;
        case "PaymentDate": szPaymentDate = this.Request.Form[szKey]; break;
        case "PaymentType": szPaymentType = this.Request.Form[szKey]; break;
        case "HandlingCharge": szHandlingCharge = this.Request.Form[szKey]; break;
        case "PaymentTypeChargeFee": szPaymentTypeChargeFee = this.Request.Form[szKey]; break;
        case "TradeDate": szTradeDate = this.Request.Form[szKey]; break;
        case "TradeStatus": szTradeStatus = this.Request.Form[szKey]; break;
        case "ItemName": szItemName = this.Request.Form[szKey]; break;
        /* 使用 WebATM 交易時回傳的參數 */
        case "WebATMAccBank": szWebATMAccBank = this.Request.Form[szKey]; break;
        case "WebATMAccNo": szWebATMAccNo = this.Request.Form[szKey]; break;
        /* 使用 ATM 交易時回傳的參數 */
        case "ATMAccBank": szATMAccBank = this.Request.Form[szKey]; break;
        case "ATMAccNo": szATMAccNo = this.Request.Form[szKey]; break;
        /* 使用 CVS 或 BARCODE 交易時回傳的參數 */
        case "PaymentNo": szPaymentNo = this.Request.Form[szKey]; break;
        case "PayFrom": szPayFrom = this.Request.Form[szKey]; break;
        /* 使用 Alipay 交易時回傳的參數 */
        case "AlipayID": szAlipayID = this.Request.Form[szKey]; break;
        case "AlipayTradeNo": szAlipayTradeNo = this.Request.Form[szKey]; break;
        /* 使用 Tenpay 交易時回傳的參數 */
        case "TenpayTradeNo": szTenpayTradeNo = this.Request.Form[szKey]; break;
        /* 使用 Credit 交易時回傳的參數 */
        case "gwsr": szGwsr = this.Request.Form[szKey]; break;
        case "process_date": szProcessDate = this.Request.Form[szKey]; break;
        case "auth_code": szAuthCode = this.Request.Form[szKey]; break;
        case "amount": szAmount = this.Request.Form[szKey]; break;
        case "stage": szStage = this.Request.Form[szKey]; break;
```

```
        case "stast": szStast = this.Request.Form[szKey]; break;
        case "staed": szStaed = this.Request.Form[szKey]; break;
        case "eci": szECI = this.Request.Form[szKey]; break;
        case "card4no": szCard4No = this.Request.Form[szKey]; break;
        case "card6no": szCard6No = this.Request.Form[szKey]; break;
        case "red_dan": szRedDan = this.Request.Form[szKey]; break;
        case "red_de_amt": szRedDeAmt = this.Request.Form[szKey]; break;
        case "red_ok_amt": szRedOkAmt = this.Request.Form[szKey]; break;
        case "red_yet": szRedYet = this.Request.Form[szKey]; break;
        case "PeriodType": szPeriodType = this.Request.Form[szKey]; break;
        case "Frequency": szFrequency = this.Request.Form[szKey]; break;
        case "ExecTimes": szExecTimes = this.Request.Form[szKey]; break;
        case "PeriodAmount": szPeriodAmount = this.Request.Form[szKey]; break;
        case "TotalSuccessTimes": szTotalSuccessTimes = this.Request.Form[szKey]; break;
        case "TotalSuccessAmount": szTotalSuccessAmount = this.Request.Form[szKey]; break;
        default: break;
    }
}
// 其他資料處理。
.....
} else {
    // 其他資料處理。
    .....
}
}
catch (Exception ex)
{
    // 例外錯誤處理。
    enErrors.Add(ex.Message);
}
finally
{
    // 顯示錯誤訊息。
    if (enErrors.Count() > 0)
        string szErrorMessage = String.Join("\r\n", enErrors);
}
```

訂單查詢程式碼範例(PHP)：

```
/*
 * 查詢訂單的範例程式碼。
 */
try
{
    $oPayment = new AllInOne();
    /* 服務參數 */
    $oPayment->ServiceURL = "<<您要呼叫的服務位址>>";
    $oPayment->HashKey = "<<AllPay提供給您的Hash Key>>";
    $oPayment->HashIV = "<<AllPay提供給您的Hash IV>>";
    $oPayment->MerchantID = "<<AllPay提供給您的特店編號>>";
    /* 基本參數 */
    $oPayment->Query['MerchantTradeNo'] = "<<您要查詢的訂單交易編號>>";
    /* 查詢訂單 */
    $arQueryFeedback = $oPayment ->QueryTradeInfo();
    // 取回所有資料
    if (sizeof($arQueryFeedback) > 0) {
        foreach ($arQueryFeedback as $key => $value) {
            switch ($key)
            {
                /* 使用 Credit 定期定額交易時回傳的參數 */
                case "MerchantID": $szMerchantID = $value; break;
                case "MerchantTradeNo": $szMerchantTradeNo = $value; break;
                case "TradeNo": $szTradeNo = $value; break;
                case "TradeAmt": $szTradeAmt = $value; break;
                case "PaymentDate": $szPaymentDate = $value; break;
                case "HandlingCharge": $szHandlingCharge = $value; break;
                case "PaymentTypeChargeFee": $szPaymentTypeChargeFee = $value; break;
                case "TradeDate": $szTradeDate = $value; break;
                case "TradeStatus": $szTradeStatus = $value; break;
                case "ItemName": $szItemName = $value; break;
                /* 使用 WebATM 交易時回傳的參數 */
                case "WebATMAccBank": $szWebATMAccBank = $value; break;
                case "WebATMAccNo": $szWebATMAccNo = $value; break;
                /* 使用 ATM 交易時回傳的參數 */
                case "ATMAccBank": $szATMAccBank = $value; break;
                case "ATMAccNo": $szATMAccNo = $value; break;
                /* 使用 CVS 或 BARCODE 交易時回傳的參數 */
            }
        }
    }
}
```

```

        case "PaymentNo": $szPaymentNo = $value; break;
        case "PayFrom": $szPayFrom = $value; break;
        /* 使用 Alipay 交易時回傳的參數 */
        case "AlipayID": $szAlipayID = $value; break;
        case "AlipayTradeNo": $szAlipayTradeNo = $value; break;
        /* 使用 Tenpay 交易時回傳的參數 */
        case "TenpayTradeNo": $szTenpayTradeNo = $value; break;
        /* 使用 Credit 交易時回傳的參數 */
        case "gwsr": $szGwsr = $value; break;
        case "process_date": $szProcessDate = $value; break;
        case "auth_code": $szAuthCode = $value; break;
        case "amount": $szAmount = $value; break;
        case "stage": $szStage = $value; break;
        case "stast": $szStast = $value; break;
        case "staed": $szStaed = $value; break;
        case "eci": $szECI = $value; break;
        case "card4no": $szCard4No = $value; break;
        case "card6no": $szCard6No = $value; break;
        case "red_dan": $szRedDan = $value; break;
        case "red_de_amt": $szRedDeAmt = $value; break;
        case "red_ok_amt": $szRedOkAmt = $value; break;
        case "red_yet": $szRedYet = $value; break;
        case "PeriodType": $szPeriodType = $value; break;
        case "Frequency": $szFrequency = $value; break;
        case "ExecTimes": $szExecTimes = $value; break;
        case "PeriodAmount": $szPeriodAmount = $value; break;
        case "TotalSuccessTimes": $szTotalSuccessTimes = $value; break;
        case "TotalSuccessAmount": $szTotalSuccessAmount = $value; break;
        default: break;
    }
}
// 其他資料處理。
.....
} else {
    // 其他資料處理。
    .....
}
}

```

```
catch (Exception $e)
{
    // 例外錯誤處理。
    .....
}
```

訂單查詢程式碼範例(JAVA)：

```
/*
 * 查詢訂單的範例程式碼。
 */
List<String> enErrors = new ArrayList<String>();
try {
    AllInOne oPayment = new AllInOne();

    /* 服務參數 */
    oPayment.ServiceMethod = HttpMethod.ServerPOST; // 或使用 HttpMethod.HttpSOAP;
    oPayment.ServiceURL = "<<您要呼叫的服務位址>>";
    oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";
    oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";
    oPayment.MerchantID = "<<AllPay提供給您的特店編號>>";

    /* 基本參數 */
    oPayment.Query.MerchantTradeNo = "<<您要查詢的訂單交易編號>>";

    Hashtable<String, String> htFeedback = new Hashtable<String, String>();

    enErrors.addAll(oPayment.QueryTradeInfo(htFeedback));

    Set<String> key = htFeedback.keySet();
    String name[] = key.toArray(new String[key.size()]);

    /* 查詢後的回傳的基本參數 */
    String szMerchantID = "";
    String szMerchantTradeNo = "";
    String szTradeNo = "";
    String szTradeAmt = "";
    String szPaymentDate = "";
    String szPaymentType = "";
```

```
String szHandlingCharge = "";
String szPaymentTypeChargeFee = "";
String szTradeDate = "";
String szTradeStatus = "";
String szItemName = "";
/* 使用 WebATM 交易時，回傳的額外參數 */
String szWebATMAccBank = "";
String szWebATMAccNo = "";
/* 使用 ATM 交易時，回傳的額外參數 */
String szATMAccBank = "";
String szATMAccNo = "";
/* 使用 CVS 或 BARCODE 交易時，回傳的額外參數 */
String szPaymentNo = "";
String szPayFrom = "";
/* 使用 Alipay 交易時，回傳的額外參數 */
String szAlipayID = "";
String szAlipayTradeNo = "";
/* 使用 Tenpay 交易時，回傳的額外參數 */
String szTenpayTradeNo = "";
/* 使用 Credit 交易時，回傳的額外參數 */
String szGwsr = "";
String szProcessDate = "";
String szAuthCode = "";
String szAmount = "";
String szStage = "";
String szStast = "";
String szStaed = "";
String szECI = "";
String szCard4No = "";
String szCard6No = "";
String szRedDan = "";
String szRedDeAmt = "";
String szRedOkAmt = "";
String szRedYet = "";
String szPeriodType = "";
String szFrequency = "";
String szExecTimes = "";
String szPeriodAmount = "";
```

```
String szTotalSuccessTimes = "";
String szTotalSuccessAmount = "";

// 取得資料
for(int i = 0 ; i < name.length ; i++) {
    /* 查詢後的回傳的基本參數 */
    if(name[i].equals("MerchantID"))
        szMerchantID = htFeedback.get(name[i]);
    else if(name[i].equals("MerchantTradeNo"))
        szMerchantTradeNo = htFeedback.get(name[i]);
    else if(name[i].equals("TradeNo"))
        szTradeNo = htFeedback.get(name[i]);
    else if(name[i].equals("TradeAmt"))
        szTradeAmt = htFeedback.get(name[i]);
    else if(name[i].equals("PaymentDate"))
        szPaymentDate = htFeedback.get(name[i]);
    else if(name[i].equals("PaymentType"))
        szPaymentType = htFeedback.get(name[i]);
    else if(name[i].equals("HandlingCharge"))
        szHandlingCharge = htFeedback.get(name[i]);
    else if(name[i].equals("PaymentTypeChargeFee"))
        szPaymentTypeChargeFee = htFeedback.get(name[i]);
    else if(name[i].equals("TradeDate"))
        szTradeDate = htFeedback.get(name[i]);
    else if(name[i].equals("TradeStatus"))
        szTradeStatus = htFeedback.get(name[i]);
    else if(name[i].equals("ItemName"))
        szItemName = htFeedback.get(name[i]);
    /* 使用 WebATM 交易時回傳的參數 */
    else if(name[i].equals("WebATMAccBank"))
        szWebATMAccBank = htFeedback.get(name[i]);
    else if(name[i].equals("WebATMAccNo"))
        szWebATMAccNo = htFeedback.get(name[i]);
    /* 使用 ATM 交易時回傳的參數 */
    else if(name[i].equals("ATMAccBank"))
        szATMAccBank = htFeedback.get(name[i]);
    else if(name[i].equals("ATMAccNo"))
        szATMAccNo = htFeedback.get(name[i]);
}
```



```

/* 使用 CVS 或 BARCODE 交易時回傳的參數 */
else if(name[i].equals("PaymentNo"))
    szPaymentNo = htFeedback.get(name[i]);
else if(name[i].equals("PayFrom"))
    szPayFrom = htFeedback.get(name[i]);
/* 使用 Alipay 交易時回傳的參數 */
else if(name[i].equals("AlipayID"))
    szAlipayID = htFeedback.get(name[i]);
else if(name[i].equals("AlipayTradeNo"))
    szAlipayTradeNo = htFeedback.get(name[i]);
/* 使用 Tenpay 交易時回傳的參數 */
else if(name[i].equals("TenpayTradeNo"))
    szTenpayTradeNo = htFeedback.get(name[i]);
/* 使用 Credit 交易時回傳的參數 */
else if(name[i].equals("gwsr"))
    szGwsr = htFeedback.get(name[i]);
else if(name[i].equals("process_date"))
    szProcessDate = htFeedback.get(name[i]);
else if(name[i].equals("auth_code"))
    szAuthCode = htFeedback.get(name[i]);
else if(name[i].equals("amount"))
    szAmount = htFeedback.get(name[i]);
else if(name[i].equals("stage"))
    szStage = htFeedback.get(name[i]);
else if(name[i].equals("stast"))
    szStast = htFeedback.get(name[i]);
else if(name[i].equals("staed"))
    szStaed = htFeedback.get(name[i]);
else if(name[i].equals("eci"))
    szECI = htFeedback.get(name[i]);
else if(name[i].equals("card4no"))
    szCard4No = htFeedback.get(name[i]);
else if(name[i].equals("card6no"))
    szCard6No = htFeedback.get(name[i]);
else if(name[i].equals("red_dan"))
    szRedDan = htFeedback.get(name[i]);
else if(name[i].equals("red_de_amt"))
    szRedDeAmt = htFeedback.get(name[i]);

```

```

        else if(name[i].equals("red_ok_amt"))
            szRedOkAmt = htFeedback.get(name[i]);
        else if(name[i].equals("red_yet"))
            szRedYet = htFeedback.get(name[i]);
        else if(name[i].equals("PeriodType"))
            szPeriodType = htFeedback.get(name[i]);
        else if(name[i].equals("Frequency"))
            szFrequency = htFeedback.get(name[i]);
        else if(name[i].equals("ExecTimes"))
            szExecTimes = htFeedback.get(name[i]);
        else if(name[i].equals("PeriodAmount"))
            szPeriodAmount = htFeedback.get(name[i]);
        else if(name[i].equals("TotalSuccessTimes"))
            szTotalSuccessTimes = htFeedback.get(name[i]);
        else if(name[i].equals("TotalSuccessAmount"))
            szTotalSuccessAmount = htFeedback.get(name[i]);
    }
    /* 查詢後的回傳的基本參數 */
    System.out.println("MerchantID = " + szMerchantID);
    System.out.println("MerchantTradeNo = " + szMerchantTradeNo);
    System.out.println("TradeNo = " + szTradeNo);
    System.out.println("TradeAmt = " + szTradeAmt);
    System.out.println("PaymentDate = " + szPaymentDate);
    System.out.println("PaymentType = " + szPaymentType);
    System.out.println("HandlingCharge = " + szHandlingCharge);
    System.out.println("PaymentTypeChargeFee = " + szPaymentTypeChargeFee);
    System.out.println("TradeDate = " + szTradeDate);
    System.out.println("TradeStatus = " + szTradeStatus);
    System.out.println("ItemName = " + szItemName);
    /* 使用 WebATM 交易時回傳的參數 */
    System.out.println("WebATMAccBank = " + szWebATMAccBank);
    System.out.println("WebATMAccNo = " + szWebATMAccNo);
    /* 使用 ATM 交易時回傳的參數 */
    System.out.println("ATMAccBank = " + szATMAccBank);
    System.out.println("ATMAccNo = " + szATMAccNo);
    /* 使用 CVS 或 BARCODE 交易時回傳的參數 */
    System.out.println("PaymentNo = " + szPaymentNo);
    System.out.println("PayFrom = " + szPayFrom);

```

```

/* 使用 Alipay 交易時回傳的參數 */
System.out.println("AlipayID = " + szAlipayID);
System.out.println("AlipayTradeNo = " + szAlipayTradeNo);
/* 使用 Tenpay 交易時回傳的參數 */
System.out.println("TenpayTradeNo = " + szTenpayTradeNo);
/* 使用 Credit 交易時回傳的參數 */
System.out.println("gwsr = " + szGwsr);
System.out.println("process_date = " + szProcessDate);
System.out.println("auth_code = " + szAuthCode);
System.out.println("amount = " + szAmount);
System.out.println("stage = " + szStage);
System.out.println("stast = " + szStast);
System.out.println("staed = " + szStaed);
System.out.println("eci = " + szECI);
System.out.println("card4no = " + szCard4No);
System.out.println("card6no = " + szCard6No);
System.out.println("red_dan = " + szRedDan);
System.out.println("red_de_amt = " + szRedDeAmt);
System.out.println("red_ok_amt = " + szRedOkAmt);
System.out.println("red_yet = " + szRedYet);
System.out.println("PeriodType = " + szPeriodType);
System.out.println("Frequency = " + szFrequency);
System.out.println("ExecTimes = " + szExecTimes);
System.out.println("PeriodAmount = " + szPeriodAmount);
System.out.println("TotalSuccessTimes = " + szTotalSuccessTimes);
System.out.println("TotalSuccessAmount = " + szTotalSuccessAmount);
out.println("");
}
catch (Exception e) {
    // 例外錯誤處理。
    enErrors.add(e.getMessage());
}
finally {
    // 顯示錯誤訊息。
    if (enErrors.size() > 0)
        out.print(enErrors);
}

```

5.1.4. 信用卡定期定額訂單查詢的方法

(QueryPeriodCreditCardTradeInfo)

本方法提供特店(賣家會員)進行訂單查詢的作業；在介接該服務之前，請至「廠商後台 > 基本資料查詢 > 廠商基本資料」設定加入特店伺服器的對外 IP 位址到「允許的 IP」欄位([請參考](#))。


介接的服務網址(正式機為 https，測試機為 http)：

Http POST

- 正式環境：<https://payment.allpay.com.tw/Cashier/QueryCreditCardTradeInfo>
- 測試環境：<http://payment-stage.allpay.com.tw/Cashier/QueryCreditCardTradeInfo>

5.1.4.1. 訂單查詢使用參數

基本資料傳送的參數成員如下表(名稱前若有紅色星號*為必填欄位)：

	名稱	型態	說明
	*Query.MerchantTradeNo	String(20)	廠商的交易編號。

接收基本回傳的參數成員如下表：

	參數	名稱	型態	說明
	MerchantID	廠商編號	String(10)	AllPay 提供的廠商編號。
	MerchantTradeNo	廠商交易編號	String(20)	產生訂單時傳送給 AllPay 的廠商交易編號。
	TradeNo	AllPay 的交易編號	String(20)	請保存 AllPay 的交易編號與 MerchantTradeNo 的關連。
	RtnCode	交易狀態	Integer	1:授權成功，其餘為失敗，失敗代碼請參考 交易狀態代碼一覽表 。
	PeriodType	週期種類	PeriodType	顯示為空、Y、M 或 D。
	Frequency	執行頻率	Integer	訂單建立時的所設定的執行頻率。
	ExecTimes	執行次數	Integer	訂單建立時的所設定的執行頻率。
	PeriodAmount	每次授權金額	Integer	訂單建立時的每次要授權金額。
	amount	授權金額	Integer	所授權的金額

	gwsr	授權交易單號	Integer	
	process_date	授權成功處理時間	String(20)	格式為 yyyy/MM/dd HH:mm:ss。
	auth_code	授權碼	String(6)	
	card4no	卡片的末 4 碼	String(4)	
	card6no	卡片的末 6 碼	String(6)	
	TotalSuccessTimes	已成功授權次數合計	Integer	目前已成功授權的次數
	TotalSuccessAmount	已成功授權總金額	Integer	目前已成功授權的金額合計
	ExecLog	執行紀錄明細		每一次的授權紀錄。 例：若已成功授權 3 次 (TotalSuccessTimes 為 3) · 則會 顯示 3 筆授權紀錄。 欄位內容詳見「執行記錄明細的 參數成員」。

執行紀錄明細的參數成員如下表：

當付款方式設定為 Credit 時				
	參數	名稱	型態	說明
	RtnCode	交易狀態	Integer	1:授權成功，其餘為失敗，失敗代碼請參考 交易狀態代碼一覽表 。
	amount	金額	Integer	
	gwsr	授權交易單號	Integer	
	process_date	處理時間	DateTime	格式為 yyyy/MM/dd HH:mm:ss。
	auth_code	授權碼	String(6)	

5.1.4.2. 定期定額訂單查詢程式碼範例

訂單查詢程式碼範例(C#)：

```

/*
 * 查詢訂單的範例程式碼。
 */
List<string> enErrors = new List<string>();
PeriodCreditCardTradeInfo oFeedback = null;

try
{
    using (AllInOne oPayment = new AllInOne())
    {

```

```
/* 服務參數 */
oPayment.ServiceMethod = HttpMethod.ServerPOST; // 或使用 HttpMethod.HttpSOAP;
oPayment.ServiceURL = "<<您要呼叫的服務位址>>";
oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";
oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";
oPayment.MerchantID = "<<AllPay提供給您的特店編號>>";
/* 基本參數 */
oPayment.Query.MerchantTradeNo = "<<您要查詢的訂單交易編號>>";
/* 查詢訂單 */
enErrors.AddRange(oPayment.QueryPeriodCreditCardTradeInfo(ref oFeedback));
}
// 取回所有資料
if (enErrors.Count() == 0)
{
    /* 查詢後的回傳的基本參數 */
    string szMerchantID = oFeedback.MerchantID;
    string szMerchantTradeNo = oFeedback.MerchantTradeNo;
    string szTradeNo = oFeedback.TradeNo;
    int nRtnCode = oFeedback.RtnCode;
    string szPeriodType = oFeedback.PeriodType;
    int nFrequency = oFeedback.Frequency;
    int nExecTimes = oFeedback.ExecTimes;
    int nPeriodAmount = oFeedback.PeriodAmount;
    int nAmount = oFeedback.amount;
    int nGwsr = oFeedback.gwsr;
    string szProcessDate = oFeedback.process_date;
    string szAuthCode = oFeedback.auth_code;
    string szCard4No = oFeedback.card4no;
    string szCard6No = oFeedback.card6no;
    int nTotalSuccessTimes = oFeedback.TotalSuccessTimes;
    int nTotalSuccessAmount = oFeedback.TotalSuccessAmount;
    // 取得明細資料
    foreach (PeriodCreditCardTradeInfoLog oLog in oFeedback.ExecLog)
    {
        /* 使用 Credit 交易時回傳的參數 */
        int nLogRtnCode = oLog.RtnCode;
        int nLogAmount = oLog.amount;
        long nLogGwsr = oLog.gwsr;
```

```
        string szLogProcessDate = oLog.process_date;
        string szLogAuthCode = oLog.auth_code;
    }
    // 其他資料處理。
    .....
} else {
    // 其他資料處理。
    .....
}
}
catch (Exception ex)
{
    // 例外錯誤處理。
    enErrors.Add(ex.Message);
}
finally
{
    // 顯示錯誤訊息。
    if (enErrors.Count() > 0)
        string szErrorMessage = String.Join("\r\n", enErrors);
}
```

訂單查詢程式碼範例(PHP)：

```
/*
 * 查詢訂單的範例程式碼。
 */
try
{
    $oPayment = new AllInOne();
    /* 服務參數 */
    $oPayment->ServiceURL = "<<您要呼叫的服務位址>>";
    $oPayment->HashKey = "<<AllPay提供給您的Hash Key>>";
    $oPayment->HashIV = "<<AllPay提供給您的Hash IV>>";
    $oPayment->MerchantID = "<<AllPay提供給您的特店編號>>";
    /* 基本參數 */
    $oPayment->Query['MerchantTradeNo'] = "<<您要查詢的訂單交易編號>>";
    /* 查詢訂單 */
    $arQueryFeedback = $oPayment ->QueryPeriodCreditCardTradeInfo();
}
```

```
// 取回所有資料
if (sizeof($arQueryFeedback) > 0) {
    $szMerchantID = $arQueryFeedback->MerchantID;
    $szMerchantTradeNo = $arQueryFeedback->MerchantTradeNo;
    $szTradeNo = $arQueryFeedback->TradeNo;
    $nRtnCode = $arQueryFeedback->RtnCode;
    $szPeriodType = $arQueryFeedback->PeriodType;
    $szFrequency = $arQueryFeedback->Frequency;
    $szExecTimes = $arQueryFeedback->ExecTimes;
    $szPeriodAmount = $arQueryFeedback->PeriodAmount;
    $szAmount = $arQueryFeedback->amount;
    $szGwsr = $arQueryFeedback->gwsr;
    $szProcessDate = $arQueryFeedback->process_date;
    $szAuthCode = $arQueryFeedback->auth_code;
    $szCard4No = $arQueryFeedback->card4no;
    $szCard6No = $arQueryFeedback->card6no;
    $szTotalSuccessTimes = $arQueryFeedback->TotalSuccessTimes;
    $szTotalSuccessAmount = $arQueryFeedback->TotalSuccessAmount;
    foreach ($arQueryFeedback->ExecLog as $key => $value) {
        switch ($key)
        {
            /* 使用 Credit 交易時回傳的參數 */
            case "RtnCode": $szLogRtnCode = $value; break;
            case "gwsr": $szLogGwsr = $value; break;
            case "process_date": $szLogProcessDate = $value; break;
            case "auth_code": $szLogAuthCode = $value; break;
            case "amount": $szLogAmount = $value; break;
        }
    }
    // 其他資料處理。
    .....
} else {
    // 其他資料處理。
    .....
}
}
catch (Exception $e)
{

```



```
// 例外錯誤處理。
```

```
.....
```

```
}
```

訂單查詢程式碼範例(JAVA)：

```
/*
 * 查詢訂單的範例程式碼。
 */
List<String> enErrors = new ArrayList<String>();
PeriodCreditCardTradeInfo oFeedback = new PeriodCreditCardTradeInfo();
try {
    AllInOne oPayment = new AllInOne();

    /* 服務參數 */
    oPayment.ServiceMethod = HttpMethod.ServerPOST; // 或使用 HttpMethod.HttpSOAP;
    oPayment.ServiceURL = "<<您要呼叫的服務位址>>";
    oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";
    oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";
    oPayment.MerchantID = "<<AllPay提供給您的特店編號>>";

    /* 基本參數 */
    oPayment.Query.MerchantTradeNo = "<<您要查詢的訂單交易編號>>";

    PeriodCreditCardTradeInfo htFeedback = new PeriodCreditCardTradeInfo();

    enErrors.addAll(oPayment.QueryPeriodCreditCardTradeInfo(htFeedback));

    /* 查詢後的回傳的基本參數 */

    String szMerchantID = htFeedback.MerchantID;
    String szMerchantTradeNo = htFeedback.MerchantTradeNo;
    String szTradeNo = htFeedback.TradeNo;
    int nRtnCode = htFeedback.RtnCode;
    String szPeriodType = htFeedback.PeriodType;
    int nFrequency = htFeedback.Frequency;
    int nExecTimes = htFeedback.ExecTimes;
    int nPeriodAmount = htFeedback.PeriodAmount;
    int nAmount = htFeedback.amount;
```

```
int nGwsr = htFeedback.gwsr;

String szProcessDate = htFeedback.process_date;

String szAuthCode = htFeedback.auth_code;

String szCard4No = htFeedback.card4no;

String szCard6No = htFeedback.card6no;

int nTotalSuccessTimes = htFeedback.TotalSuccessTimes;

int nTotalSuccessAmount = htFeedback.TotalSuccessAmount;

PeriodCreditCardTradeInfoLog oLog[] = htFeedback.ExecLog;

//

System.out.println("MerchantID = " + szMerchantID);

System.out.println("MerchantTradeNo = " + szMerchantTradeNo);

System.out.println("TradeNo = " + szTradeNo);

System.out.println("RtnCode = " + nRtnCode);

System.out.println("PeriodType = " + szPeriodType);

System.out.println("Frequency = " + nFrequency);

System.out.println("ExecTimes = " + nExecTimes);

System.out.println("PeriodAmount = " + nPeriodAmount);

System.out.println("amount = " + nAmount);

System.out.println("gwsr = " + nGwsr);

System.out.println("process_date = " + szProcessDate);

System.out.println("auth_code = " + szAuthCode);

System.out.println("card4no = " + szCard4No);

System.out.println("card6no = " + szCard6No);

System.out.println("TotalSuccessTimes = " + nTotalSuccessTimes);

System.out.println("TotalSuccessAmount = " + nTotalSuccessAmount);

// 取得明細資料

if(nTotalSuccessTimes > 0) {

    for(int i = 0 ; i < oLog.length ; i++) {

        int nLogRtnCode = oLog[i].RtnCode;

        int nLogAmount = oLog[i].amount;

        long nLogGwsr = oLog[i].gwsr;

        String szLogProcessDate = oLog[i].process_date;

        String szLogAuthCode = oLog[i].auth_code;

        System.out.println("count [" + i + "] RtnCode = " + nLogRtnCode);

        System.out.println("count [" + i + "] amount = " + nLogAmount);

    }

}
```

```
        System.out.println("count [" + i + "] gwsr = " + nLogGwsr);
        System.out.println("count [" + i + "] process_date = " + szLogProcessDate);
        System.out.println("count [" + i + "] auth_code = " + szLogAuthCode);
    }
}
}
catch (Exception e) {
    // 例外錯誤處理。
    enErrors.add(e.getMessage());
}
finally {
    // 顯示錯誤訊息。
    if (enErrors.size() > 0)
        out.print(enErrors);
}
```

5.1.5. 信用卡關帳/退刷/取消/放棄的方法(DoAction)

當您的接收程式收到訂單授權成功後，即可以執行此 API 對歐付寶發動「關帳」、「退刷」、「取消」、「放棄」的動作。

關帳說明：當您的接收程式收到授權成功後，即可以利用此 API 對歐付寶進行關帳的動作。

退刷說明：對已經有請款動作之授權交易，才能作退刷的動作(分期的交易一定得全額退刷，一般授權才能作部份金額退刷)。

取消說明：此一功能的使用時機是，當您有發動關帳或退款，在當然晚上 8:30 分未送銀行請款之前，都還可以取消該請款或退款的動作(此一取消之發動，會將該天所有該筆授權要關帳中的帳目不論請款或退款，都取消掉喔)。





放棄說明：此一功能的使用時機，必須是該筆授權都還沒有任何請款關帳記錄，才能進行(也就是放棄此筆交易，不作請款)。

介接的服務網址(正式機為 https，測試機為 http)：

- 正式環境：<https://payment.allpay.com.tw/CreditDetail/DoAction>
- 測試環境：<http://payment-stage.allpay.com.tw/CreditDetail/DoAction>

5.1.5.1. 信用卡訂單處理使用參數

基本參數成員如下表(名稱前若有紅色星號*為必填欄位)：

	名稱	型態	說明
	*Action.MerchantTradeNo	String(20)	廠商的交易編號。
	*Action.TradeNo	String(20)	請保存 AllPay 的交易編號與 MerchantTradeNo 的關連。
	*Action.Action	ActionType	針對訂單做處理的動作。 (C：關帳、R：退刷、E：取消、N：放棄)
	*Action.TotalAmount	Decimal	訂單總金額。

基本回傳的參數成員如下表：

	參數	名稱	型態	說明
	MerchantID	廠商編號	String(10)	AllPay 提供的廠商編號。
	MerchantTradeNo	廠商交易編號	String(20)	產生訂單時傳送給 AllPay 的廠商交易編號。
	TradeNo	AllPay 的交易編號	String(20)	請保存 AllPay 的交易編號與 MerchantTradeNo 的關連。
	RtnCode	交易狀態	Integer	1 表示成功，其餘為失敗，請參考 交易訊息代碼一覽表 。
	RtnMsg	商品名稱	String(200)	

5.1.5.2. 信用卡訂單處理程式碼範例

信用卡訂單處理程式碼範例(C#)：

```
/*
 * 信用卡關帳/退刷/取消/放棄的範例程式碼。
 */
List<string> enErrors = new List<string>();
Hashtable htFeedback = null;

try
{
    using (AllInOne oPayment = new AllInOne())
    {
        /* 服務參數 */
        oPayment.ServiceMethod = HttpMethod.ServerPOST;
```

```
oPayment.ServiceURL = "<<您要呼叫的服務位址>>";
oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";
oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";
oPayment.MerchantID = "<<AllPay提供給您的特店編號>>";
/* 基本參數 */
oPayment.Action.MerchantTradeNo = "<<您要處理的訂單交易編號>>";
oPayment.Action.TradeNo = "<<AllPay的交易編號>>";
oPayment.Action.Action = ActionType.C;
oPayment.Action.TotalAmount = Decimal.Parse("<<訂單總金額>>");

enErrors.AddRange(oPayment.DoAction(ref htFeedback));
}

if (enErrors.Count() == 0)
{
    /* 執行後的回傳的基本參數 */
    string szMerchantID = String.Empty;
    string szMerchantTradeNo = String.Empty;
    string szTradeNo = String.Empty;
    string szRtnCode = String.Empty;
    string szRtnMsg = String.Empty;
    // 取得資料於畫面
    foreach (string szKey in htFeedback.Keys)
    {
        switch (szKey)
        {
            /* 執行後的回傳的基本參數 */
            case "MerchantID": szMerchantID = this.Request.Form[szKey]; break;
            case "MerchantTradeNo": szMerchantTradeNo = this.Request.Form[szKey]; break;
            case "TradeNo": szTradeNo = this.Request.Form[szKey]; break;
            case "RtnCode": szRtnCode = this.Request.Form[szKey]; break;
            case "RtnMsg": szRtnMsg = this.Request.Form[szKey]; break;
            default: break;
        }
    }
    // 其他資料處理。
    .....
} else {
```

```

        // 其他資料處理。
        .....
    }
}
catch (Exception ex)
{
    // 例外錯誤處理。
    enErrors.Add(ex.Message);
}
finally
{
    // 顯示錯誤訊息。
    if (enErrors.Count() > 0)
        string szErrorMessage = String.Join("\r\n", enErrors);
}

```

信用卡訂單處理程式碼範例(PHP)：

```

/*
 * 信用卡關帳/退刷/取消/放棄的範例程式碼。
 */
try
{
    $oPayment = new AllInOne();
    /* 服務參數 */
    $oPayment->ServiceURL = "<<您要呼叫的服務位址>>";
    $oPayment->HashKey = "<<AllPay提供給您的Hash Key>>";
    $oPayment->HashIV = "<<AllPay提供給您的Hash IV>>";
    $oPayment->MerchantID = "<<AllPay提供給您的特店編號>>";
    /* 基本參數 */
    $oPayment->Action['MerchantTradeNo'] = "<<您要處理的訂單交易編號>>";
    $oPayment->Action['TradeNo'] = "<<AllPay的交易編號>>";
    $oPayment->Action['Action'] = ActionType::C;
    $oPayment->Action['TotalAmount'] = (int)"<<訂單總金額>>";

    $sFeedback = $oPayment->DoAction();
}

```

```
if (sizeof($arFeedback) > 0) {
    foreach ($arFeedback as $key => $value) {
        switch ($key)
        {
            /* 使用 Credit 定期定額交易時回傳的參數 */
            case "MerchantID": $szMerchantID = $value; break;
            case "MerchantTradeNo": $szMerchantTradeNo = $value; break;
            case "TradeNo": $szTradeNo = $value; break;
            case "RtnCode": $szTradeAmt = $value; break;
            case "RtnMsg": $szPaymentDate = $value; break;
            default: break;
        }
    }
    // 其他資料處理。
    .....
} else {
    // 其他資料處理。
    .....
}
}
catch (Exception $e)
{
    // 例外錯誤處理。
    .....
}
```

信用卡訂單處理程式碼範例(JAVA)：

```
/*
 * 信用卡關帳/退刷/取消/放棄的範例程式碼。
 */
List<String> enErrors = new ArrayList<String>();
try {
    AllInOne oPayment = new AllInOne();

    /* 服務參數 */
    oPayment.ServiceMethod = HttpMethod.ServerPOST;
    oPayment.ServiceURL = "<<您要呼叫的服務位址>>";
    oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";
```

```
oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";
oPayment.MerchantID = "<<AllPay提供給您的特店編號>>";

/* 基本參數 */
oPayment.Action.MerchantTradeNo = "<<您要處理的訂單交易編號>>";
oPayment.Action.TradeNo = "<<AllPay的交易編號>>";
oPayment.Action.Action = ActionType.C;
oPayment.Action.TotalAmount = new Decimal("<<訂單總金額>>");

Hashtable<String, String> htFeedback = new Hashtable<String, String>();

enErrors.addAll(oPayment.DoAction(htFeedback));

Set<String> key = htFeedback.keySet();
String name[] = key.toArray(new String[key.size()]);

/* 執行後的回傳的基本參數 */
String szMerchantID = "";
String szMerchantTradeNo = "";
String szTradeNo = "";
String szRtnCode = "";
String szRtnMsg = "";

// 取得資料
for(int i = 0 ; i < name.length ; i++) {
    /* 執行後的回傳的基本參數 */
    if(name[i].equals("MerchantID"))
        szMerchantID = htFeedback.get(name[i]);
    else if(name[i].equals("MerchantTradeNo"))
        szMerchantTradeNo = htFeedback.get(name[i]);
    else if(name[i].equals("TradeNo"))
        szTradeNo = htFeedback.get(name[i]);
    else if(name[i].equals("RtnCode"))
        szRtnCode = htFeedback.get(name[i]);
    else if(name[i].equals("RtnMsg"))
        szRtnMsg = htFeedback.get(name[i]);
}

/* 執行後的回傳的基本參數 */
```



```

        System.out.println("MerchantID = " + szMerchantID);
        System.out.println("MerchantTradeNo = " + szMerchantTradeNo);
        System.out.println("TradeNo = " + szTradeNo);
        System.out.println("RtnCode = " + szRtnCode);
        System.out.println("RtnMsg = " + szRtnMsg);
    }
    catch (Exception e) {
        // 例外錯誤處理。
        enErrors.add(e.getMessage());
    }
    finally {
        // 顯示錯誤訊息。
        if (enErrors.size() > 0)
            out.print(enErrors);
    }
}
    
```

5.1.6. 廠商通知退款的方法(AioChargeback)





特店(賣家會員)可利用此 API 將消費金額退回買家的歐付寶帳戶。

介接的服務網址(正式機為 https，測試機為 http)：

- 正式環境：<https://payment.allpay.com.tw/Cashier/AioChargeback>
- 測試環境：<http://payment-stage.allpay.com.tw/Cashier/AioChargeback>

5.1.6.1. 廠商通知退款使用參數

基本資料傳送的參數成員如下表(名稱前若有紅色星號*為必填欄位)：

	名稱	型態	說明
	*ChargeBack.MerchantTradeNo	String(20)	廠商的交易編號。
	*ChargeBack.TradeNo	String(20)	請保存 AllPay 的交易編號與 MerchantTradeNo 的關連。
	*ChargeBack.ChargeBackTotalAmount	Decimal	訂單總金額。
	ChargeBack.Remark	String(100)	備註欄位，目前都請放空白。

基本回傳的參數成員如下表：

參數	名稱	型態	說明
----	----	----	----

	RtnCode	交易狀態	Integer	1 表示成功，其餘為失敗，請參考 交易訊息代碼一覽表 。
	RtnMsg	商品名稱	String(200)	

5.1.6.2. 廠商通知退款程式碼範例

廠商通知退款程式碼範例(C#)：

```

/*
 * 廠商通知退款的範例程式碼。
 */
List<string> enErrors = new List<string>();
Hashtable htFeedback = null;

try
{
    using (AllInOne oPayment = new AllInOne())
    {
        /* 服務參數 */
        oPayment.ServiceMethod = HttpMethod.ServerPOST;
        oPayment.ServiceURL = "<<您要呼叫的服務位址>>";
        oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";
        oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";
        oPayment.MerchantID = "<<AllPay提供給您的特店編號>>";
        /* 基本參數 */
        oPayment.ChargeBack.MerchantTradeNo = "<<您要退款的訂單交易編號>>";
        oPayment.ChargeBack.TradeNo = "<<AllPay的交易編號>>";
        oPayment.ChargeBack.Remark = "<<您要填寫的其他備註>>";
        oPayment.ChargeBack.ChargeBackTotalAmount = Decimal.Parse("<<退款總金額>>");

        enErrors.AddRange(oPayment.AioChargeback(ref htFeedback));
    }

    if (enErrors.Count() == 0)
    {
        /* 執行後的回傳的基本參數 */
        string szRtnCode = String.Empty;
        string szRtnMsg = String.Empty;
        // 取得資料於畫面
        foreach (string szKey in htFeedback.Keys)

```

```

        {
            switch (szKey)
            {
                /* 執行後的回傳的基本參數 */
                case "RtnCode": szRtnCode = this.Request.Form[szKey]; break;
                case "RtnMsg": szRtnMsg = this.Request.Form[szKey]; break;
                default: break;
            }
        }
        // 其他資料處理。
        .....
    } else {
        // 其他資料處理。
        .....
    }
}
catch (Exception ex)
{
    // 例外錯誤處理。
    enErrors.Add(ex.Message);
}
finally
{
    // 顯示錯誤訊息。
    if (enErrors.Count() > 0)
        string szErrorMessage = String.Join("\r\n", enErrors);
}

```

廠商通知退款程式碼範例(PHP)：

```

/*
 * 廠商通知退款的範例程式碼。
 */
try
{
    $oPayment = new AllInOne();
    /* 服務參數 */
    $oPayment->ServiceURL = "<<您要呼叫的服務位址>>";
    $oPayment->HashKey = "<<AllPay提供給您的Hash Key>>";
}

```

```

$Payment->HashIV = "<<AllPay提供給您的Hash IV>>";
$Payment->MerchantID = "<<AllPay提供給您的特店編號>>";
/* 基本參數 */
$Payment->ChargeBack['MerchantTradeNo'] = "<<您要退款的訂單交易編號>>";
$Payment->ChargeBack['TradeNo'] = "<<AllPay的交易編號>>";
$Payment->ChargeBack['Remark'] = "<<您要填寫的其他備註>>";
$Payment->ChargeBack['ChargeBackTotalAmount'] = (int)"<<退款總金額>>";

$arResult = $Payment->AioChargeback();

if (sizeof($arResult) > 0) {
    foreach ($arResult as $key => $value) {
        switch ($key)
        {
            case "RtnCode": $szTradeAmt = $value; break;
            case "RtnMsg": $szPaymentDate = $value; break;
            default: break;
        }
    }
    // 其他資料處理。
    .....
} else {
    // 其他資料處理。
    .....
}
}
catch (Exception $e)
{
    // 例外錯誤處理。
    .....
}

```

廠商通知退款程式碼範例(JAVA)：

```

/*
 * 廠商通知退款的範例程式碼。
 */
List<String> enErrors = new ArrayList<String>();
try {

```

```
AllInOne oPayment = new AllInOne();

/* 服務參數 */
oPayment.ServiceMethod = HttpMethod.ServerPOST;
oPayment.ServiceURL = "<<您要呼叫的服務位址>>";
oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";
oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";
oPayment.MerchantID = "<<AllPay提供給您的特店編號>>";

/* 基本參數 */
oPayment.ChargeBack.MerchantTradeNo = "<<您要退款的訂單交易編號>>";
oPayment.ChargeBack.TradeNo = "<<AllPay的交易編號>>";
oPayment.ChargeBack.Remark = "<<您要填寫的其他備註>>";
oPayment.ChargeBack.ChargeBackTotalAmount = new Decimal("<<退款總金額>>");

Hashtable<String, String> htFeedback = new Hashtable<String, String>();

enErrors.addAll(oPayment.AioChargeback(htFeedback));
Set<String> key = htFeedback.keySet();
String name[] = key.toArray(new String[key.size()]);

/* 執行後的回傳的基本參數 */
String szRtnCode = "";
String szRtnMsg = "";




// 取得資料
for(int i = 0 ; i < name.length ; i++) {
    /* 執行後的回傳的基本參數 */
    if(name[i].equals("RtnCode"))
        szRtnCode = htFeedback.get(name[i]);
    else if(name[i].equals("RtnMsg"))
        szRtnMsg = htFeedback.get(name[i]);
}

/* 執行後的回傳的基本參數 */
System.out.println("RtnCode = " + szRtnCode);
System.out.println("RtnMsg = " + szRtnMsg);
}
catch (Exception e) {
```

```
// 例外錯誤處理。
enErrors.add(e.getMessage());
}
finally {
    // 顯示錯誤訊息。
    if (enErrors.size() > 0)
        out.print(enErrors);
}
```

5.2. 全功能有履約保證類別(AllInEscrow Class)

該類別提供全功能有履約的介接方法，其資料傳送的參數成員如下表(名稱前若有紅色星號*為必填欄位)：

	名稱	型態	說明
	*ServiceURL	String(200)	要呼叫介接服務的網址。
	*ServiceMethod	HttpMethod	介接服務時，呼叫 API 的方法，預設為 HttpPOST。
	*HashKey	String	AllPay 提供的 HashKey。
	*HashIV	String	AllPay 提供的 HashIV。
	*MerchantID	String(10)	AllPay 提供的廠商編號。
	Send	SendArguments	介接產生訂單方法時，所需要的參數內容。
	Delivery	DeliveryArguments	介接出貨通知方法時，所需要的參數內容。
	Change	OrderChangeArgunments	介接商品異常/取消訂單時，所需要參數內容。
	Query	QueryArguments	介接訂單查詢時，所需要參數內容。
	CheckOut	IEnumerable<String>	產生訂單的方法(不支援 Ajax 非同步)。
	CheckOut(String)	IEnumerable<String>	於指定視窗產生訂單的方法(不支援 Ajax 非同步)。
	CheckOutFeedback	IEnumerable<String>	取回訂單處理結果的方法。
	DeliveryNotify	IEnumerable<String>	出貨通知的方法。
	OrderChangeFeedback	IEnumerable<String>	接收 AllPay 通知商品異常/取消訂單的方法。
	OrderChangeNotify	IEnumerable<String>	商品異常/取消訂單處理結果回覆的方法。
	QueryTradeInfo	IEnumerable<String>	查詢訂單資料的方法。

建立該類別的範例如下：

```
/*
 * 全功能介接的建構範例程式碼。
 */
```

```
try
{
    using (AllInEscrow oPayment = new AllInEscrow())
    {
        /* 服務參數 */
        oPayment.ServiceMethod = HttpMethod.HttpPOST;
        oPayment.ServiceURL = "<<您要呼叫的服務位址>>";
        oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";
        oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";
        oPayment.MerchantID = "<<AllPay提供給您的特店編號>>";
        .....
    }
}
catch (Exception ex)
{
    // 例外錯誤處理。
}
finally
{
    // 顯示錯誤訊息。
}
```

5.2.1. 訂單產生的方法(CheckOut)

該方法僅適用於 Web 站台或 Web 專案，提供您使用 HTTP POST 的方式，於瀏覽器端送出訂單資料到 AllPay 的支付介面，進行訂單的付款。例：使用者在特店(賣家會員)消費後，特店(賣家會員)呼叫該方法，該方法會自動產生 POST 頁面，將訂單資料送至 AllPay 進行付款結帳作業，結帳完成後，AllPay 會將結果通知特店(賣家會員)，特店(賣家會員)可於接收頁面呼叫 [CheckOutFeedback 方法](#)，取回付款處理結果。

介接的服務網址(正式機為 https，測試機為 http)：

- 正式環境：<https://payment.allpay.com.tw/Cashier/CheckOut>
- 測試環境：<http://payment-stage.allpay.com.tw/Cashier/CheckOut>

基本資料傳送的參數成員如下表(名稱前若有紅色星號*為必填欄位)：

名稱	型態	說明
----	----	----

	*Send.ReturnURL	String(200)	付款完成通知回傳的網址。
	Send.ClientBackURL	String(200)	瀏覽器端返回的廠商網址。 ※頁面導回的時候，不會帶付款結果到此網址，只是將頁面導回而已。
	*Send.MerchantTradeNo	String(20)	廠商的交易編號，該交易編號不可重複。 ※該編號不可重複，所以不建議為訂單編號，建議產生該筆訂單要付款時使用的交易編號。
	*Send.MerchantTradeDate	DateTime	廠商的交易時間。
	*Send.TotalAmount	Decimal	交易總金額。
	*Send.TradeDesc	String(200)	交易描述。
	Send.Currency	String(5)	交易幣別，目前僅接受 TWD。
	Send.EncodeCharset	String(10)	中文編碼格式(Big5/utf-8)，預設為 utf-8。
	Send.UseAllpayAddress	Boolean	是否採用 AllPay 平台提供的住址。(預設 False)
	Send.CreditInstallment	Integer	刷卡分期期數，不分期時為 0。(預設 0)
	Send.InstallmentAmount	Decimal	刷卡分期的付款金額，不分期時為 0。(預設 0)
	Send.Redeem	Boolean	是否使用信用卡紅利折抵。(預設 False)
	Send.ShippingDate	String(8)	預計出貨日。(格式：yyyyMMdd)
	Send.ConsiderHour	Integer	商敏猶豫期時間，單位為小時，請勿低於 168 小時。(預設 168)
	Send.Remark	String(100)	備註欄位。
	*Send.Items	ItemCollection< Item >	訂單的商品資料。

訂單商品資料傳送的成員如下表(名稱前若有紅色星號*為必填欄位)

	名稱	型態	說明
	*Name	String(20)	商品名稱。
	*Price	Decimal	商品單價。
	*Currency	String(10)	幣別單位。(例如：元)
	*Quantity	Integer	購買數量。
	URL	String(200)	商品的說明網址。

產生訂單範例：

```

/*
 * 產生訂單的範例程式碼。
 */
List<string> enErrors = new List<string>();

try

```



```
{
    using (AllInEscrow oPayment = new AllInEscrow())
    {
        /* 服務參數 */
        oPayment.ServiceMethod = HttpMethod.HttpPOST;
        oPayment.ServiceURL = "<<您要呼叫的服務位址>>";
        oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";
        oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";
        oPayment.MerchantID = "<<AllPay提供給您的特店編號>>";
        /* 基本參數 */
        oPayment.Send.ReturnURL = "<<您要收到付款完成通知的伺服器端網址>>";
        oPayment.Send.ClientBackURL = "<<您要歐付寶返回按鈕導向的瀏覽器端網址>>";
        oPayment.Send.MerchantTradeNo = "<<您此筆訂單交易編號>>";
        oPayment.Send.MerchantTradeDate = DateTime.Now("<<您此筆訂單的交易時間>>");
        oPayment.Send.TotalAmount = Decimal.Parse("<<您此筆訂單的交易總金額>>");
        oPayment.Send.TradeDesc = "<<您該筆訂單的描述>>";
        oPayment.Send.Currency = "TWD";
        oPayment.Send.EncodeCharset = "utf-8";
        oPayment.Send.UseAllpayAddress = Boolean.Parse("<<您是否採用AllPay平台提供的住址>>");
        oPayment.Send.CreditInstallment = Int32.Parse("<<您的刷卡分期數>>");
        oPayment.Send.InstallmentAmount = Decimal.Parse("<<您使用刷卡分期的付款金額>>");
        oPayment.Send.Redeem = Boolean.Parse("<<是否使用紅利折抵>>");
        oPayment.Send.ShippingDate = "<<您的預計出貨日>>";
        oPayment.Send.ConsiderHour = Int32.Parse("<<您的商品猶豫期時間>>");
        oPayment.Send.Remark = "<<您要填寫的其他備註>>";
        // 加入選購商品資料。
        oPayment.Send.Items.Add(new Item() {
            Name = "<<產品A>>", Price = Decimal.Parse("<<單價>>"), Currency = "<<幣別>>",
            Quantity = Int32.Parse("<<數量>>"), URL = "<<產品說明位址>>"
        });
        oPayment.Send.Items.Add(new Item() {
            Name = "<<產品B>>", Price = Decimal.Parse("<<單價>>"), Currency = "<<幣別>>",
            Quantity = Int32.Parse("<<數量>>"), URL = "<<產品說明位址>>"
        });
        oPayment.Send.Items.Add(new Item() {
            Name = "<<產品C>>", Price = Decimal.Parse("<<單價>>"), Currency = "<<幣別>>",
            Quantity = Int32.Parse("<<數量>>"), URL = "<<產品說明位址>>"
        });
    }
}
```

```
        enErrors.AddRange(oPayment.CheckOut());
    }
}
catch (Exception ex)
{
    // 例外錯誤處理。
    enErrors.Add(ex.Message);
}
finally
{
    // 顯示錯誤訊息。
    if (enErrors.Count() > 0)
        ScriptManager.RegisterStartupScript(this, typeof(Page), "_MESSAGE", String.Format("alert('{0}');",
String.Join("\r\n", enErrors)), true);
}
```

5.2.2. 取得付款結果通知的方法(CheckOutFeedback)

當 AllPay 會員付款後，或是訂單建立成功後，AllPay 會以 Server POST 的方式傳送處理結果至產生訂單時指定的 ReturnURL 位址，請於收到訊息後，回應已接收訊息。

回應訊息格式為(註 1)：




成功：1|OK

錯誤：0|<<您的錯誤訊息內容>>

※注意事項:

註 1：若沒有收到正確訊息，系統會隔三分鐘後重發訊息給特店(賣家會員)，三次後則會延至隔天再發訊息。

接收一般付款結果通知的參數成員如下表：

參數	名稱	型態	說明
 MerchantID	廠商編號	String(10)	AllPay 提供的廠商編號。
 MerchantTradeNo	廠商交易編號	String(20)	產生訂單時傳送給 AllPay 的廠商交易編號。
 RtnCode	交易狀態	Integer	1 表示付款成功，800 表示貨到付款訂單建立成功，其餘代碼為

				失敗(請參考 交易訊息代碼一覽表)。
	RtnMsg	交易訊息	String(200)	
	TradeNo	AllPay 的交易編號	String(20)	請保存 AllPay 的交易編號與 MerchantTradeNo 的關連。
	TradeAmt	交易金額	Decimal	若歐付寶會員選擇信用卡分期付款時，此交易金額會回傳商家在訂單產生時，InstallmentAmount 所帶的金額。
	PaymentDate	付款時間	DateTime	格式為 yyyy/MM/dd HH:mm:ss。
	PaymentType	會員選擇的付款方式	PaymentMethodItem	
	PaymentType ChargeFee	通路費	Decimal	
	TradeDate	訂單成立時間	DateTime	格式為 yyyy/MM/dd HH:mm:ss。
	UserAddress	使用者要寄送的地址	String(200)	
	SimulatePaid	是否為模擬付款	Integer	1：模擬付款 0：非模擬付款 賣家會員可透過廠商後台網站的履約保證訂單管理來針對單筆訂單模擬由歐付寶發動付款通知，以方便介接 API 的測試(註 1)。

※注意事項：

註 1：當 SimulatePaid 的值為 1 時，RtnCode 也會是 1，表示此筆訂單是模擬付款，並非是由消費者實際真的付款，所以歐付寶也不會撥款給廠商，請勿對該筆交易做出貨等動作，以避免損失。

一般付款結果通知範例：

```

/*
 * 接收訂單資料產生完成的範例程式碼。
 */
List<string> enErrors = new List<string>();
Hashtable htFeedback = null;

try
{
    using (AllInEscrow oPayment = new AllInEscrow())
    {

```

```
oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";
oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";
/* 取回付款結果 */
enErrors.AddRange(oPayment.CheckOutFeedback(ref htFeedback));
}
// 取回所有資料
if (enErrors.Count() == 0)
{
    /* 支付後的回傳的基本參數 */
    string szMerchantID = String.Empty;
    string szMerchantTradeNo = String.Empty;
    string szPaymentDate = String.Empty;
    string szPaymentType = String.Empty;
    string szPaymentTypeChargeFee = String.Empty;
    string szRtnCode = String.Empty;
    string szRtnMsg = String.Empty;
    string szSimulatePaid = String.Empty;
    string szTradeAmt = String.Empty;
    string szTradeDate = String.Empty;
    string szTradeNo = String.Empty;
    // 取得資料
    foreach (string szKey in htFeedback.Keys)
    {
        switch (szKey)
        {
            /* 支付後的回傳的基本參數 */
            case "MerchantID": szMerchantID = this.Request.Form[szKey]; break;
            case "MerchantTradeNo": szMerchantTradeNo = this.Request.Form[szKey]; break;
            case "PaymentDate": szPaymentDate = this.Request.Form[szKey]; break;
            case "PaymentType": szPaymentType = this.Request.Form[szKey]; break;
            case "PaymentTypeChargeFee": szPaymentTypeChargeFee = this.Request.Form[szKey]; break;
            case "RtnCode": szRtnCode = this.Request.Form[szKey]; break;
            case "RtnMsg": szRtnMsg = this.Request.Form[szKey]; break;
            case "SimulatePaid": szSimulatePaid = this.Request.Form[szKey]; break;
            case "TradeAmt": szTradeAmt = this.Request.Form[szKey]; break;
            case "TradeDate": szTradeDate = this.Request.Form[szKey]; break;
            case "TradeNo": szTradeNo = this.Request.Form[szKey]; break;
            case "UserAddress": szUserAddress = this.Request.Form[szKey]; break;
```

```

        default: break;
    }
}
// 其他資料處理。
.....
}
}
catch (Exception ex)
{
    // 例外錯誤處理。
    enErrors.Add(ex.Message);
}
finally
{
    this.Response.Clear();
    // 回覆成功訊息。
    if (enErrors.Count() == 0)
        this.Response.Write("1|OK");
    // 回覆錯誤訊息。
    else
        this.Response.Write(String.Format("0|{0}", String.Join("\r\n", enErrors)));
    this.Response.Flush();
    this.Response.End();
}

```

5.2.3. 通知出貨的方法(DeliveryNotify)

當合作特店(賣家會員)收到 AllPay 的付款結果通知後，該訂單的商品有下列狀況時，需以 Server POST 的方式通知 AllPay 出貨的結果：

- (1) 當商品進行出貨時，需通知 AllPay 該筆訂單已出貨。
- (2) 當商品因無貨可出等其它原因要取消出貨時，需要將已付款的訂單退款給買家，則需通知 AllPay 取消該筆訂單，退款給買家。

介接的服務網址(正式機為 https，測試機為 http)：

- 正式環境：<https://payment.allpay.com.tw/Cashier/DeliveryNotify>
- 測試環境：<http://payment-stage.allpay.com.tw/Cashier/DeliveryNotify>

基本資料傳送的參數成員如下表(名稱前若有紅色星號*為必填欄位)：

	名稱	型態	說明
	*Delivery.MerchantTradeNo	String(20)	廠商的交易編號。
	*Delivery.TradeNo	String(20)	請保存 AllPay 的交易編號與 MerchantTradeNo 的關連。
	*Delivery.ShippingState	ShippingState	出貨狀態。
	Delivery.ShippingDate	DateTime	出貨時間。
	Delivery.EncodeChartset	String(10)	中文編碼格式(Big5/utf-8) · 預設為 utf-8。
	Delivery.Remark	String(100)	備註欄位。

基本回傳的參數成員如下表：

	參數	名稱	型態	說明
	RtnCode	交易狀態	Integer	1 表示成功 · 其餘為失敗 · 請參考 交易訊息代碼一覽表 。
	RtnMsg	商品名稱	String(200)	

廠商通知出貨程式碼範例：

```
/*
 * 出貨通知的範例程式碼。
 */
List<string> enErrors = new List<string>();
Hashtable htFeedback = null;

try
{
    using (AllInEscrow oPayment = new AllInEscrow())
    {
        /* 服務參數 */
        oPayment.ServiceMethod = HttpMethod.ServerPOST;
        oPayment.ServiceURL = "<<您要呼叫的服務位址>>";
        oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";
        oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";
        oPayment.MerchantID = "<<AllPay提供給您的特店編號>>";
        /* 基本參數 */
        oPayment.Delivery.MerchantTradeNo = "<<您的訂單交易編號>>";
        oPayment.Delivery.TradeNo = "<<AllPay的交易編號>>";
        oPayment.Delivery.ShippingState = ShippingState.Delivery;
```

```

        oPayment.Delivery.ShippingDate = DateTime.Now;
        oPayment.Delivery.EncodeChartset = "utf-8";
        oPayment.Delivery.Remark = "<<您要填寫的其他備註>>";

        enErrors.AddRange(oPayment.Delivery.Notify(ref htFeedback));
    }

    if (enErrors.Count() == 0)
    {
        // 其他資料處理。
        .....
    }
}
catch (Exception ex)
{
    // 例外錯誤處理。
    enErrors.Add(ex.Message);
}
finally
{
    // 顯示錯誤訊息。
    if (enErrors.Count() > 0)
        ScriptManager.RegisterStartupScript(this, typeof(Page), "_MESSAGE", String.Format("alert('{0}')");",
String.Join("\r\n", enErrors)), true);
}

```

5.2.4. 接收商品異常/取消訂單通知的方法(OrderChangeFeedback)

交易履約期間，如果消費者在 AllPay 官網進行下列作業：

- (1) 商品異常通知(退/換貨)。
- (2) 取消訂單作業(未出貨狀態下)。

則 AllPay 會以 Server POST 方式傳送資料參數通知特店(賣家會員)。

特店(賣家會員)在收到通知後，請直接在網頁回應以下格式訊息：

成功：1|OK

錯誤：0|<<您的錯誤訊息內容>>

並於五天內通知 AllPay 處理結果，通知格式[請參考下一章節](#)說明。

接收通知的參數成員如下表：

參數	名稱	型態	說明
 MerchantID	廠商編號	String(10)	AllPay 提供的廠商編號。
 MerchantTradeNo	廠商交易編號	String(20)	產生訂單時傳送給 AllPay 的廠商交易編號。
 TradeNo	AllPay 的交易編號	String(20)	請保存 AllPay 的交易編號與 MerchantTradeNo 的關連。
 InfoType	通知種類	String(3)	<p>3：取消訂單(未出貨狀態下) 4：商品異常</p> <p>當合作特店收到「3-取消訂單」時，若此時商品已經出貨，請 Call 通知出貨的方法。</p> <p>若收到「4-商品異常」時，請依照 下一章節 的說明處理。</p>
 Reason	通知理由	String(200)	
 Remark	備註欄位	String(100)	目前都請放空值。

接收商品異常/取消訂單通知範例：

```

/*
 * 商品異常/取消訂單的範例程式碼。
 */
List<string> enErrors = new List<string>();
Hashtable htFeedback = null;

try
{
    using (AllInEscrow oPayment = new AllInEscrow())
    {
        oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";
        oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";
        /* 取回通知結果 */
        enErrors.AddRange(oPayment.CheckOutFeedback(ref htFeedback));
    }
    // 取回所有資料
    if (enErrors.Count() == 0)

```



```
{
    /* 收到通知後取回的基本參數 */
    string szMerchantID = String.Empty;
    string szMerchantTradeNo = String.Empty;
    string szTradeNo = String.Empty;
    string szInfoType = String.Empty;
    string szReason = String.Empty;
    string szRemark = String.Empty;
    // 取得資料於畫面
    foreach (string szKey in htFeedback.Keys)
    {
        switch (szKey)
        {
            /* 收到通知後取回的基本參數 */
            case "MerchantID": szMerchantID = this.Request.Form[szKey]; break;
            case "MerchantTradeNo": szMerchantTradeNo = this.Request.Form[szKey]; break;
            case "InfoType": szInfoType = this.Request.Form[szKey]; break;
            case "Reason": szReason = this.Request.Form[szKey]; break;
            case "Remark": szRemark = this.Request.Form[szKey]; break;
            case "TradeNo": szTradeNo = this.Request.Form[szKey]; break;
            default: break;
        }
    }
    // 其他資料處理。
    .....
}
catch (Exception ex)
{
    enErrors.Add(ex.Message);
}
finally
{
    this.Response.Clear();
    // 回覆成功訊息。
    if (enErrors.Count() == 0)
    {
        this.Response.Write("1|OK");
    }
    // 回覆錯誤訊息。
}
```

```
else
    this.Response.Write(String.Format("0|{0}", String.Join("\r\n", enErrors)));
this.Response.Flush();
this.Response.End();
}
```

5.2.5. 商品異常/取消訂單處理結果回覆的方法(OrderChangeNotify)

當 AllPay 通知特店(賣家會員) 商品異常/取消訂單後，請於五天內通知 AllPay 該筆訂單的處理結果；或是消費者直接於特店(賣家會員)的平台進行退貨/換貨/取消訂單時，請將訂單的處理結果以 Server POST 方式通知 AllPay。

介接的服務網址(正式機為 https，測試機為 http)：

- 正式環境：<https://payment.allpay.com.tw/Cashier/OrderChangeNotify>
- 測試環境：<http://payment-stage.allpay.com.tw/Cashier/OrderChangeNotify>

基本資料傳送的參數成員如下表(名稱前若有紅色星號*為必填欄位)：

	名稱	型態	說明
	*Change.MerchantTradeNo	String(20)	廠商的交易編號。
	*Change.TradeNo	String(20)	請保存 AllPay 的交易編號與 MerchantTradeNo 的關連。
	*Change.InfoType	NotifiedType	通知類型。(註 1)
	Change.RefundAmount	Decimal	退款金額。(註 2)
	Change.Remark	String(100)	備註欄位。

基本回傳的參數成員如下表：

	參數	名稱	型態	說明
	RtnCode	交易狀態	Integer	1 表示成功，其餘為失敗，請參考 交易訊息代碼一覽表 。
	RtnMsg	商品名稱	String(200)	

※注意事項：

註 1：通知類型 InfoType 參數用途。

(1) 當值為 RefundPartAmount 時，表示要退原始訂單金額中的部份款項給消費者，則 AllPay 會需要再經過消費者同意後，才會將退款金額退給消費者。

(2) 當值為 RefundAllAmount 時，表示要退原始訂單金額的所有款項。

(3) 當值為 **Redelivery** 時，表示合作特店已確認過換貨的商品無誤，重新出貨給消費者，此時履約期會重新計算。

(4) 當消費者有可能誤按了商品異常通知，或是經合作特店詢問消費者後，商品無任何問題，無需退款或重新出貨，則請帶 **GoodsNoAbnormality**。此時履約期不會重新計算。

註 2：當通知類型 **InfoType** 為 **RefundPartAmount** 時，退款金額 **RefundAmount** 才須填寫金額，且金額請勿超過原始訂單金額。

商品異常/取消訂單處理結果回覆的範例：

```
/*
 * 商品異常/取消訂單處理結果回覆的範例程式碼。
 */

List<string> enErrors = new List<string>();
Hashtable htFeedback = null;

try
{
    using (AllInEscrow oPayment = new AllInEscrow())
    {
        /* 服務參數 */
        oPayment.ServiceMethod = HttpMethod.ServerPOST;
        oPayment.ServiceURL = "<<您要呼叫的服務位址>>";
        oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";
        oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";
        oPayment.MerchantID = "<<AllPay提供給您的特店編號>>";
        /* 基本參數 */
        oPayment.Change.MerchantTradeNo = "<<您的訂單交易編號>>";
        oPayment.Change.TradeNo = "<<AllPay的交易編號>>";
        oPayment.Change.InfoType = NotifiedType.GoodsNoAbnormality;
        oPayment.Change.RefundAmount = (String.IsNullOrEmpty("<<您的退款金額>>")) ? null :
(Decimal?)Decimal.Parse("<<您的退款金額>>");
        oPayment.Change.Remark = "<<您要填寫的其他備註>>";

        enErrors.AddRange(oPayment.OrderChangeNotify(ref htFeedback));
    }

    if (enErrors.Count() == 0)
    {
        // 其他資料處理。
    }
}
```

```
.....  
    }  
}  
catch (Exception ex)  
{  
    enErrors.Add(ex.Message);  
}  
finally  
{  
    if (enErrors.Count() > 0)  
    {  
        ScriptManager.RegisterStartupScript(this, typeof(Page), "_MESSAGE", String.Format("alert('{0}')",  
String.Join("\\r\\n", enErrors)), true);  
    }  
}
```

5.2.6. 接收訂單履約完成通知的方法(CompletedFeedback)

當 AllPay 會員的訂單履約完成時，則 AllPay 會以 Server POST 方式發送履約完成的訊息給賣家，商家必須在「廠商後台 > 基本資料查詢 > 廠商基本資料」設定一般商品履約完成網址(請參考[前置準備事項](#))，以便接收 AllPay 的通知訊息，並請於收到訊息後，回應已接收訊息。回應訊息格式為：

成功：1|OK

錯誤：0|<<您的錯誤訊息內容>>

※注意事項:

註 1：若沒有收到正確訊息，系統會隔三分鐘後重發訊息給特店(賣家會員)，三次後則會延至隔天再發訊息。

註 2：除了回應訊息外，請勿有任何 HTML 標籤、空格等其它符號。

接收通知的參數成員如下表：

	參數	名稱	型態	說明
	MerchantID	廠商編號	String(10)	AllPay 提供的廠商編號。
	MerchantTradeNo	廠商交易編號	String(20)	產生訂單時傳送給 AllPay 的廠商交易編號。
	TradeNo	AllPay 的交易編號	String(20)	請保存 AllPay 的交易編號與

				MerchantTradeNo 的關連。
	TradeStatus	交易狀態	String(3)	<p>3：取消訂單(未出貨狀態下)</p> <p>4：商品異常</p> <p>當合作特店收到「3-取消訂單」時，若此時商品已經出貨，請 Call 通知出貨的方法。</p> <p>若收到「4-商品異常」時，請依照 下一章節 的說明處理。</p>
	TotalAmount	交易金額	Decimal	
	RefundAmount	退款金額	Decimal	
	EscrowCompletedDate	履約完成時間	DateTime	格式為 yyyy/MM/dd HH:mm:ss。

接收訂單履約完成通知的範例：

```

/*
 * 接收訂單履約完成通知資料的範例程式碼。
 */
List<string> enErrors = new List<string>();
Hashtable htFeedback = null;

try
{
    using (AllInEscrow oPayment = new AllInEscrow())
    {
        oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";
        oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";
        /* 取回履約完成結果 */
        enErrors.AddRange(oPayment.CheckOutFeedback(ref htFeedback));
    }
    // 取回所有資料
    if (enErrors.Count() == 0)
    {
        /* 收到通知後取回的基本參數 */
        string szMerchantID = String.Empty;
        string szMerchantTradeNo = String.Empty;
        string szTradeNo = String.Empty;
        string szTradeStatus = String.Empty;
    }
}

```

```
string szTotalAmount = String.Empty;
string szReundAmount = String.Empty;
string szEscrowCompletedDate = String.Empty;
// 取得資料於畫面
foreach (string szKey in htFeedback.Keys)
{
    switch (szKey)
    {
        /* 收到通知後取回的基本參數 */
        case "MerchantID": szMerchantID = this.Request.Form[szKey]; break;
        case "MerchantTradeNo": szMerchantTradeNo = this.Request.Form[szKey]; break;
        case "TradeStatus": szTradeStatus = this.Request.Form[szKey]; break;
        case "TotalAmount": szTotalAmount = this.Request.Form[szKey]; break;
        case "ReundAmount": szReundAmount = this.Request.Form[szKey]; break;
        case "TradeNo": szTradeNo = this.Request.Form[szKey]; break;
        case "EscrowCompletedDate": szEscrowCompletedDate = this.Request.Form[szKey]; break;
        default: break;
    }
}
// 其他資料處理。
}
}
catch (Exception ex)
{
    enErrors.Add(ex.Message);
}
finally
{
    this.Response.Clear();
    // 回覆成功訊息。
    if (enErrors.Count() == 0)
        this.Response.Write("I|OK");
    // 回覆錯誤訊息。
    else
        this.Response.Write(String.Format("0|{0}", String.Join("\r\n", enErrors)));
    this.Response.Flush();
    this.Response.End();
}
```

5.2.7. 訂單查詢的方法(CheckOut)

本方法提供特店(賣家會員)進行訂單查詢的作業；在介接該服務之前，請至「廠商後台 > 基本資料查詢 > 廠商基本資料」設定加入特店伺服器的對外 IP 位址到「允許的 IP」欄位([請參考](#))。

介接的服務網址(正式機為 https，測試機為 http)：


Http POST

- 正式環境：<https://payment.allpay.com.tw/Cashier/QueryTradeInfo>
- 測試環境：<http://payment-stage.allpay.com.tw/Cashier/QueryTradeInfo>

Http SOAP(Web Service)





- 正式環境：<https://payment.allpay.com.tw/api/purchase/allpay.asmx?op=QueryTradeInfo>
- 測試環境：<http://payment-stage.allpay.com.tw/api/purchase/allpay.asmx?op=QueryTradeInfo>

基本資料傳送的參數成員如下表(名稱前若有紅色星號*為必填欄位)：

	名稱	型態	說明
	*Query.MerchantTradeNo	String(20)	廠商的交易編號。

接收基本回傳的參數成員如下表：

	參數	名稱	型態	說明
	MerchantID	廠商編號	String(10)	AllPay 提供的廠商編號。
	MerchantTradeNo	廠商交易編號	String(20)	產生訂單時傳送給 AllPay 的廠商交易編號。
	TradeNo	AllPay 的交易編號	String(20)	請保存 AllPay 的交易編號與 MerchantTradeNo 的關連。
	TradeAmt	交易金額	Decimal	若歐付寶會員選擇信用卡分期付款時，此交易金額會回傳商家在訂單產生時，InstallmentAmount 所帶的金額。
	PaymentDate	付款時間	DateTime	格式為 yyyy/MM/dd HH:mm:ss。
	PaymentType	會員選擇的付款方式	PaymentMethodItem	
	HandlingCharge	手續費合計	Decimal	
	PaymentType ChargeFee	通路費	Decimal	
	TradeDate	訂單成立時間	DateTime	格式為 yyyy/MM/dd HH:mm:ss。

	TradeStatus	交易狀態	Integer	請參考 交易狀態代碼一覽表 。
	ItemName	商品名稱	String(200)	
	ConsiderHour	商品猶豫期時間	Integer	商品猶豫期時間，單位為小時。
	EscrowCompleted Date	履約完成時間	DateTime	格式為 yyyy/MM/dd HH:mm:ss。

訂單查詢程式碼範例：

```

/*
 * 查詢訂單的範例程式碼。
 */
List<string> enErrors = new List<string>();
Hashtable htFeedback = null;

try
{
    using (AllInEscrow oPayment = new AllInEscrow())
    {
        /* 服務參數 */
        oPayment.ServiceMethod = HttpMethod.ServerPOST; // 或使用 HttpMethod.HttpSOAP;
        oPayment.ServiceURL = "<<您要呼叫的服務位址>>";
        oPayment.HashKey = "<<AllPay提供給您的Hash Key>>";
        oPayment.HashIV = "<<AllPay提供給您的Hash IV>>";
        oPayment.MerchantID = "<<AllPay提供給您的特店編號>>";
        /* 基本參數 */
        oPayment.Query.MerchantTradeNo = "<<您要查詢的訂單交易編號>>";
        /* 查詢訂單 */
        enErrors.AddRange(oPayment.QueryTradeInfo(ref htFeedback));
    }
    // 取回所有資料
    if (enErrors.Count() == 0)
    {
        /* 查詢後的回傳的基本參數 */
        string szMerchantID = String.Empty;
        string szMerchantTradeNo = String.Empty;
        string szTradeNo = String.Empty;
        string szTradeAmt = String.Empty;
        string szPaymentDate = String.Empty;
        string szPaymentType = String.Empty;
    }
}

```



```
string szHandlingCharge = String.Empty;
string szPaymentTypeChargeFee = String.Empty;
string szTradeDate = String.Empty;
string szTradeStatus = String.Empty;
string szItemName = String.Empty;
string szConsiderHour = String.Empty;
string szEscrowCompleted = String.Empty;
// 取得資料於畫面
foreach (string szKey in htFeedback.Keys)
{
    switch (szKey)
    {
        /* 查詢後的回傳的基本參數 */
        case "MerchantID": szMerchantID = this.Request.Form[szKey]; break;
        case "MerchantTradeNo": szMerchantTradeNo = this.Request.Form[szKey]; break;
        case "TradeNo": szTradeNo = this.Request.Form[szKey]; break;
        case "TradeAmt": szTradeAmt = this.Request.Form[szKey]; break;
        case "PaymentDate": szPaymentDate = this.Request.Form[szKey]; break;
        case "PaymentType": szPaymentType = this.Request.Form[szKey]; break;
        case "HandlingCharge": szHandlingCharge = this.Request.Form[szKey]; break;
        case "PaymentTypeChargeFee": szPaymentTypeChargeFee = this.Request.Form[szKey]; break;
        case "TradeDate": szTradeDate = this.Request.Form[szKey]; break;
        case "TradeStatus": szTradeStatus = this.Request.Form[szKey]; break;
        case "ItemName": szItemName = this.Request.Form[szKey]; break;
        case "ConsiderHour": szConsiderHour = this.Request.Form[szKey]; break;
        case "EscrowCompleted": szEscrowCompleted = this.Request.Form[szKey]; break;
        default: break;
    }
}
// 其他資料處理。
.....
}
}
catch (Exception ex)
{
    // 例外錯誤處理。
    enErrors.Add(ex.Message);
}
```

```
finally
{
    // 顯示錯誤訊息。
    if (enErrors.Count() > 0)
        ScriptManager.RegisterStartupScript(this, typeof(Page), "_MESSAGE",
            String.Format("alert('{0}');", String.Join("\r\n", enErrors)), true);
}
```

6. 參考資料

文件名稱	下載位址
AllPay 全功能金流整合介接規格說明	下載
AllPay 履約保證介接規格說明	下載





7. 附錄

7.1. 呼叫服務的方法(HttpMethod)




























	名稱	值	說明
	HttpGet	0	以 HTTP 的 GET 方法，於瀏覽器端傳遞資料。
	HttpPost	1	以 HTTP 的 POST 方法，於瀏覽器端傳遞資料。
	ServerPOST	2	以 HTTP 的 POST 方法，於伺服器端傳遞資料。
	HttpSOAP	3	以 SOAP(Web Service)方法，於伺服器端傳遞資料。



7.2. 付款方式(PaymentMethod)

	名稱	值	說明
	ALL	0	不指定付款方式。
	Credit	1	信用卡付費。
	WebATM	2	網路 ATM。
	ATM	3	自動櫃員機。
	CVS	4	超商代碼。



	BARCODE	5	超商條碼。
	Alipay	6	支付寶。
	Tenpay	7	財付通。
	TopUpUsed	8	儲值消費。

7.3. 付款方式子項目(PaymentMethodItem)



	名稱	值	說明
	None	0	不指定。
	WebATM_TAISHIN	1	台新銀行。
	WebATM_ESUN	2	玉山銀行。
	WebATM_HUANAN	3	華南銀行。
	WebATM_BOT	4	台灣銀行。
	WebATM_FUBON	5	台北富邦。
	WebATM_CHINATRUST	6	中國信託。
	WebATM_FIRST	7	第一銀行。
	WebATM_CATHAY	8	國泰世華。
	WebATM_MEGA	9	兆豐銀行。
	WebATM_YUANTA	10	元大銀行。
	WebATM_LAND	11	土地銀行。
	ATM_TAISHIN	101	台新銀行。
	ATM_ESUN	102	玉山銀行。
	ATM_HUANAN	103	華南銀行。
	ATM_BOT	104	台灣銀行。
	ATM_FUBON	105	台北富邦。
	ATM_CHINATRUST	106	中國信託。
	ATM_FIRST	107	第一銀行。
	CVS	201	超商代碼繳款。
	CVS_OK	202	OK 超商代碼繳款。
	CVS_FAMILY	203	全家超商代碼繳款。
	CVS_HILIFE	204	萊爾富超商代碼繳款。
	CVS_IBON	205	7-11 ibon 代碼繳款。
	Alipay	311	支付寶。
	Tenpay	321	財付通。
	TopUpUsed_AllPay	401	儲值/餘額消費(歐付寶)
	TopUpUsed_ESUN	402	儲值/餘額消費(玉山)
	BARCODE	901	超商條碼繳款。

	Credit	911	信用卡(MasterCard/JCB/VISA)。
	COD	921	貨到付款。





7.4. 額外付款資訊(ExtraPaymentInfo)

	名稱	值	說明
	Yes	1	需要額外付款資訊。
	No	0	不需要額外付款資訊。

7.5. 裝置類型(DeviceType)

	名稱	值	說明
	PC	0	桌機版付費頁面。
	Mobile	1	行動裝置版付費頁面。

7.6. 定期定額的週期種類(PeriodType)





	名稱	值	說明
	None		無。
	Year	Y	年。
	Month	M	月。
	Day	D	日。

7.7. 信用卡訂單處理動作(ActionType)



	名稱	值	說明
	C	0	關帳。
	R	1	退刷。
	E	2	取消。
	N	3	放棄。

7.8. 訂單異動通知類型(NotifiedType)



	名稱	值	說明
--	----	---	----

	RefundPartAmount	1	退回部分金額。
	RefundAllAmount	2	退回全部金額。
	Redelivery	3	換貨後重新出貨。
	GoodsNoAbnormality	4	商品無異常。

7.9. 出貨狀態(ShippingState)

	名稱	值	說明
	Delivery	5	出貨。
	Cancel	6	取消出貨(當為無貨可出等原因需要取消出貨)。



7.10. 電子發票開立註記(InvoiceState)

	名稱	值	說明
	Yes	Y	需要開立電子發票。
	No		不需要開立電子發票。

7.11. 電子發票載具類別(CarruerType)



	名稱	值	說明
	None		無載具。
	Member	1	會員載具。
	Citizen	2	買受人自然人憑證。
	Cellphone	3	買受人手機條碼。

7.12. 電子發票捐贈註記(Donation)



	名稱	值	說明
	Yes	1	捐贈。
	No	2	不捐贈(預設值)。

7.13. 電子發票列印註記(PrintMark)


	名稱	值	說明
--	----	---	----

	No	0	不列印(預設值)。
	Yes	1	列印。



7.14. 電子發票課稅類別(TaxType)

	名稱	值	說明
	Dutiable	1	應稅。
	Zero	2	零稅率。
	Free	3	免稅。
	Mix	9	應稅與免稅混合(限收銀機發票無法分辨時使用,且需通過申請核可)。

7.15. 電子發票通關方式(ClearanceMark)

	名稱	值	說明
	Yes	1	經海關出口。
	No	2	非經海關出口。

7.16. 電子發票字軌類別(InvType)

	名稱	值	說明
	General	07	一般稅額。
	Special	08	特種稅額。

7.17. 交易訊息代碼一覽表

因錯誤代碼一直在新增，詳細的錯誤代碼，請到「廠商後台->系統開發管理->交易狀態代碼查詢」功能查詢。

交易代碼	交易訊息	中文訊息
1	Succeeded	成功
2	Create Trade Succeeded.	建立訂單成功

800	COD Create Succeeded.	貨到付款訂單建立成功
10200001	Can not use trade service.	無法使用此服務
10200002	Trade has been updated before.	交易已處理
10200003	Trade Status Error.	交易狀態錯誤
10200005	Price Format Error.	金額格式錯誤
10200007	ItemURL Format Error.	URL 格式錯誤
10200050	AllPayTradeID Error.	AllPay 交易編號錯誤
10200051	MerchantID Error.	廠商編號錯誤
10200052	MerchantTradeNo Error.	廠商交易編號錯誤
10100001	IP Access Denied.	IP 拒絕存取
10100054	Trading Number Repeated.	交易編號重覆
10100073		取號成功
10200073	CheckMacValue Error	檢查碼錯誤
10100055	Create Trade Fail.	建立訂單失敗
10100058	Pay Fail.	付款失敗
10100059	Trading Number cannot Be Found.	找不到訂單編號

7.18. 交易狀態代碼一覽表

交易代碼	交易訊息	中文訊息
0	TradeSetUpState	訂單成立，等待付款
1	履約交易完成	履約交易完成
100	已付款	已付款
110	等待退款給買家	等待退款給買家
111	等待匯款賣家	等待匯款給賣家
112	出貨取消，等待退款	出貨取消，等待退款
113	訂單取消退款	訂單取消退款
114	重新出貨準備中	重新出貨準備中
127	等待買家同意退款	等待買家同意退款
2	退款完成	退款完成
3	部份退款完成	部份退款完成
300	已出貨	已出貨
302	商品異常等待處理	商品異常等待處理
340	已付款交易取消中	已付款交易取消中
370	通知賣家取消訂單	通知賣家取消訂單
700	客服處理中	客服處理中

