

智付寶股份有限公司

信用卡-請退款模組 技術串接手冊

標準版

文件版本號：V 1.0.3

版本異動

異動序號	文件版本號	修改內容	日期
1	V 1.0.0	初版	2014/7/30
2	V1.0.1	增加分期付款規則	2014/9/3
3	V1.0.2	增加紅利扣抵規則	2016/3/29
4	V1.0.3	附件一新增.net c#版本程式語言編寫 AES 加密 語法參考	2016/5/9

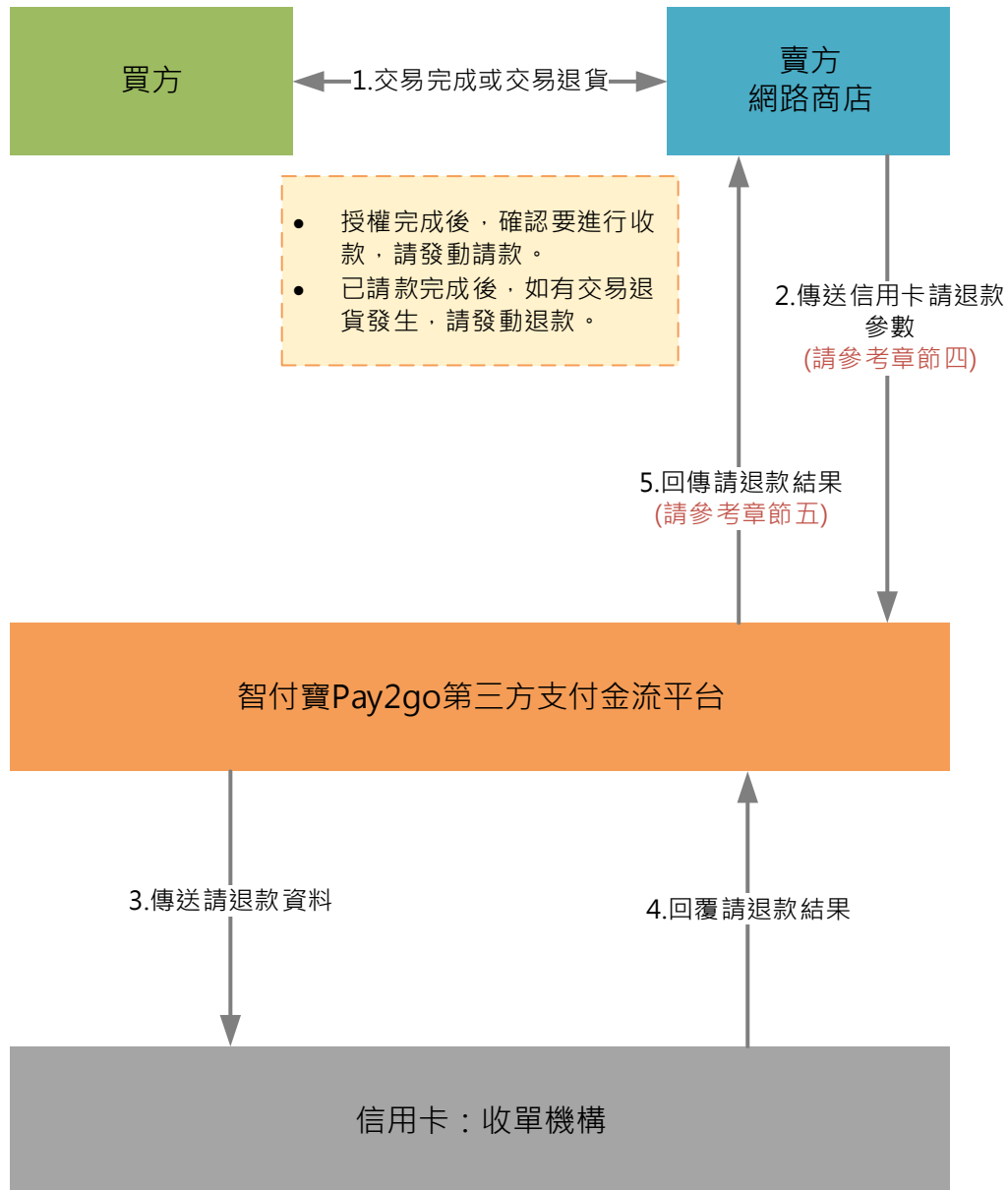
目 錄

一、串接目的.....	3
二、作業流程圖.....	4
三、串接環境與作業流程.....	5
四、Gateway 信用卡請退款參數設定.....	8
五、系統回應訊息.....	10
六、錯誤代碼.....	12
附件一 PostData_加密方法.....	13

一、串接目的

1. 商店將已完成授權的交易訊息傳送至智付寶，由智付寶向收單機構發動請款，並在接收回應訊息時立即回覆商店，以利商店後續作業。
2. 商店將已完成請款的交易訊息傳送至智付寶，由智付寶向收單機構發動退款，並在接收回應訊息時立即回覆商店，以利商店後續作業。

二、作業流程圖



三、串接環境與作業流程

(一) 測試串接網址：<https://cweb.pay2go.com/API/CreditCard/Close>

1. 請於智付寶測試平台註冊會員並建立測試商店。
 - (1) 請至智付寶測試平台 <https://cweb.pay2go.com/>於頁面右上方點選「註冊」按鈕。
 - (2) 填寫資料註冊成為會員並依平台導引建立測試商店。
2. 取得 API 串接金鑰 Hash Key 及 Hash IV。
 - (1) 請登入智付寶測試平台，於【銷售中心】單元點選目錄【商店管理/商店資料設定】。
 - (2) 點選「詳細資料」，就可取得商店 API 串接金鑰 Hash Key 及 Hash IV。
3. 於測試環境完成串接後，進行交易測試，步驟如下：
 - (1) 建立交易訂單
 - (2) 選擇支付方式及測試付款

代碼	中文名稱	測試交易注意事項
CREDIT	信用卡	1. 測試網站僅接受以下的測試卡號： 4000-2222-2222-2222 (一次付清與分期付款) 4003-5522-2222-2222 (紅利折抵) 2. 測試卡號有效月年及卡片背面末三碼，請任意填寫。 3. 系統在執行測試刷卡後，以測試授權碼回應模擬付款完成。 4. 以測試卡號之外的卡號資料進行交易都會失敗。

4. 測試交易查詢：於【銷售中心】單元點選目錄【銷售記錄查詢】進行查看。

【備註】：您於智付寶測試平台註冊的會員帳號密碼及測試資料，有效使用天期為註冊完成日起算 30 天。

(二) 正式串接網址：<https://web.pay2go.com/API/CreditCard/Close>

1. 請於智付寶平台註冊會員並建立商店。

- (1) 請至智付寶平台 <https://www.pay2go.com/>於頁面右上方點選「註冊」按鈕。
 - (2) 填寫資料註冊成為會員並依平台導引建立商店。
 2. 取得 API 串接金鑰 Hash Key 及 Hash IV。
 - (1) 請登入智付寶平台，於【銷售中心】單元點選目錄【商店管理/商店資料設定】。
 - (2) 點選「詳細資料」，就可取得商店 API 串接金鑰 Hash Key 及 Hash IV。
 3. 正式開始使用智付寶平台服務。
- (三) 信用卡申請請款執行條件：
1. 申請請款交易須為授權完成狀態。
 2. 請款有效期限為授權成功日起算 21 個日曆日晚上九點前。
 3. 一次付清交易每筆請款金額必須小於或等於授權金額，不可重覆請款。
 4. 分期付款交易每筆請款金額必須等於授權金額，不可重覆請款。
 5. 若該筆訂單以紅利折抵方式進行交易，則請款金額以訂單金額為準，每筆請款金額必須等於訂單金額，不可重覆請款。
- 例如：
- (1) 訂單金額為 NT\$1,000，顧客使用信用卡紅利折抵進行支付，銀行進行紅利折抵後，回覆授權金額為 NT\$900，請款金額必須為訂單金額 NT\$1,000。
 - (2) 訂單金額為 NT\$1,000，顧客使用信用卡紅利折抵進行支付，但顧客紅利不足，銀行進行紅利折抵後，回覆授權金額為 NT\$1,000，則請款金額必須為 NT\$1,000，雖未實際扣除紅利，但仍視為信用卡紅利折抵交易。
- (四) 信用卡申請退款執行條件：
1. 申請退款交易須為已請款狀態

2. 退款有效期限為請款日起算 90 個日曆日晚上九點前。(款項認列時間以收單機構為準。)
3. 一次付清交易每筆退款金額必須小於或等於請款金額，退款次數以每日一次為限，當日不可重覆退款。
4. 分期付款交易每筆退款金額必須等於請款金額，不可部份退款。
5. 若該筆訂單以紅利折抵方式進行交易，則每筆退款金額必須等於請款金額，不可部份退款。

例如：

- (1) 訂單金額為 NT\$1,000，顧客使用信用卡紅利折抵進行支付，銀行進行紅利折抵後，回覆授權金額為 NT\$900，請款金額為訂單金額 NT\$1,000，退款金額也必須為 NT\$1,000。
- (2) 訂單金額為 NT\$1,000，顧客使用信用卡紅利折抵進行支付，但顧客紅利不足，銀行進行紅利折抵後，回覆授權金額為 NT\$1,000，請款金額為訂單金額 NT\$1,000，退款金額也必須為 NT\$1,000，未實際扣除紅利，但仍視為信用卡紅利折抵交易。

(五) 資料交換方式

1. 商店以幕後「HTTP POST」方式傳送請退款資料至智付寶平台進行請退款。智付寶依約定時間以批次寫入方式，將請退款資料送至收單機構。
2. 傳送方式須以標準的 Form Post 進行。
3. 智付寶將以 Web Services 方式回應。
4. 編碼格式為 UTF-8。

四、信用卡請退款參數設定

(一) Post 參數說明：

參數名稱	參數中文名稱	必填	型態	備註
MerchantID_	商店代號	V	Varchar(15)	智付寶商店代號。
PostData_	加密資料	V	text	相關加密方法請參考“ 附件一 ”

註：參數 MerchantID_及 PostData_ 後方有底線”_”符號。

(二) PostData_內含欄位：

參數名稱	參數中文名稱	必填	型態	備註
RespondType	回傳格式	V	Varchar(5)	String 或是 JSON
Version	串接程式版本	V	Varchar(5)	請帶 1.0。
Amt	請退款金額	V	Int(10)	1.純數字不含符號 2.一次刷卡請款金額需小於或等於授權金額 3. 一次刷卡退款金額需小於或等於請款金額 4.分期付款請款金額須等於授權金額 5.分期付款退款金額須等於請款金額 6. 若該筆訂單以紅利折抵方式進行交易，則請款金額以訂單金額為準，每筆請款金額必須等於訂單金額 7. 若該筆訂單以紅利折抵方式進行交易，則每筆退款金額必須等於請款金額
MerchantOrderNo	商店訂單編號	V	Varchar(20)	1.同一商店中此編號不可重覆 2.只接受英文或數字與底線
TimeStamp	時間戳記	V	Varchar(30)	自從 Unix 紀元 (格林威治時間 1970 年 1 月 1 日 00:00:00)到當前時間的秒數，若以 php 程式語言為例，即為呼叫 time()函式所回傳的值 Ex. 2014-05-15 15:00:00 這個時間的時間戳記為 1400137200
IndexType	選用單號類別	V	Int(1)	1.只能填數字 1 或 2

				1 代表選用商店訂單編號 2 代表選用智付寶交易序號 2.當選用其中一種單號類別時，該種單號不可空白
TradeNo	智付寶交易序號	V	Varchar(20)	智付寶平台交易序號
CloseType	請款或退款	V	Int(1)	請款交易時請填 1，退款交易時請填 2

五、系統回應訊息

(一) 智付寶平台接收到商店請退款資訊後，回應接收結果並顯示於 WebServices 頁面。

(二) 回應訊息格式為依 RespondType 所帶的參數回傳對應格式。

回傳格式為兩種：Json 及 String：

1. 若 RespondType = JSON，則請參考以下「JSON 回傳參數說明」。
2. 若 RespondType = String，則請參考以下「String 回傳參數說明」。

(1) JSON 回傳參數說明：

請以 Form Post 方式來接 JSONData 變數，取得以下 JSON 格式的參數資料。

參數名稱	參數中文名稱	型態	備註
Status	回傳狀態	Varchar(10)	1.若請退款成功則回傳 SUCCESS。 2.若失敗則回傳錯誤代碼。 錯誤代碼請參考“ 六、錯誤代碼 ”
Message	回傳訊息	Varchar(30)	文字，敘述此次請退款訊息
Result	回傳資料	JSON	內容格式為 JSON

上述 Result 欄位內含下列參數，並以陣列儲存後以 JSON 編碼後回應，商店接收後需使用 JSON 解碼方式後讀取使用。

以 PHP 程式語言為例，可參考 JSON_ENCODE()與 JSON_DECODE 函式。

參數名稱	參數中文名稱	型態	備註
MerchantID	商店代號	Varchar(15)	商店代號
Amt	請退款金額	Int(15)	本次交易請退款金額
TradeNo	智付寶交易序號	Varchar(20)	智付寶平台交易序號
MerchantOrderNo	商店訂單編號	Varchar(20)	請退款的商店訂單編號

範例程式：

```

{"Status":"TRA10027","Message":"\u6b64\u8a02\u55ae\u5df2\u7533\u8acb\u904e\u8acb\u6b3e\u0c\u4e0d\u53ef\u91cd\u8986\u8acb\u6b3e","Result":{"MerchantID":"11250","Amt":"10","MerchantOrderNo":"20140519193443"}}
    
```

(2) String 回傳參數說明：

參數名稱	參數中文名稱	型態	備註
Status	回傳狀態	Varchar(10)	1.若請退款成功則回傳 SUCCESS。 2.若失敗則回傳錯誤代碼。 錯誤代碼請參考 " 六、錯誤代碼 "
Message	回傳訊息	Varchar(30)	文字，敘述此次請退款訊息
MerchantID	商店代號	Varchar(15)	商店代號
Amt	交易金額	Int(15)	本次交易請退款金額
TradeNo	智付寶交易序號	Varchar(20)	智付寶平台交易序號
MerchantOrderNo	商店訂單編號	Varchar(20)	請退款的商店訂單編號

範例程式：

```
Status=TRA10027&Message=此訂單已申請過請款，不可重覆請款
&MerchantID=11250&Amt=10&MerchantOrderNo=20140519193443
```

六、錯誤代碼

商店傳送刷卡資料請退款時，可能遭遇下列錯誤狀況，其錯誤代碼與對應的錯誤原因如下表所示：

錯誤代碼	錯誤原因	備註
MEM40012	資料傳遞錯誤_PostData 空白	
MEM40013	資料缺少	會提示空白欄位
MEM40008	資料空白	會提示空白欄位
MEM40014	傳送時間錯誤	
TRA10001	商店代號錯誤	
TRA10003	金額填入非數字資訊	
TRA10008	資料加密錯誤	
TRA10009	商店代號空白	
TRA10012	商店代號停用	
TRA10013	信用卡資格停用	
TRA10045	該筆交易今日已退款	
TRA10058	分期交易非全額請款或退款	
TRA10675	紅利交易只接受全額請退款	
TRA20004	商店訂單編號重覆	
TRA20011	該筆交易已請款	

附件一 postData_加密方法

A. postData_加密方法(以 PHP 為例)：

使用此 Gateway 串接規格時，商店傳送的交易資料中，除了商店代號欄位(MerchantID_)之外，其餘欄位均須採用 Aes256 加密後放到 postData_欄位，再以 PHP 版本程式語言透過幕後方式送至信用卡請款閘道，可參考 CURL 模組。

以下提供 PHP 版本程式語言範例以供參考：

```
<?php

$url = "https://cweb.pay2go.com/API/CreditCard/Close";

$merchant_id = "123456"; //商店代號

$key = "tZT07t9z5PxxMVN1YBJRtqzhaaZJo1pS"; //商家專屬加密用 key 值

$iv = "1jT8N2HSSfoFacH8"; //商家專屬加密用 iv 值

$ResponseType= "JSON"; // 可為 JSON 或 String

//$TradeNo = "14053012102338869";
$MerchantOrderNo = "1401423320";

//====以下為副程式====

function addpadding($string, $blocksize = 32) {
    $len = strlen($string);
    $pad = $blocksize - ($len % $blocksize);
    $string .= str_repeat(chr($pad), $pad);
    return $string;
}

function curl_work($url = "", $parameter = "") {
    $curl_options = array(
        CURLOPT_URL => $url,
        CURLOPT_HEADER => false,
        CURLOPT_RETURNTRANSFER => true,
```

```

        CURLOPT_USERAGENT => "Google Bot",
//      CURLOPT_FOLLOWLOCATION => true,
        CURLOPT_SSL_VERIFYPEER => FALSE,
        CURLOPT_SSL_VERIFYHOST => FALSE,
        CURLOPT_POST => "1",
        CURLOPT_POSTFIELDS => $parameter
    );
    $ch = curl_init();
curl_setopt_array($ch, $curl_options);
    $result = curl_exec($ch);
    $retcode = curl_getinfo($ch, CURLINFO_HTTP_CODE);
    $curl_error = curl_errno($ch);
curl_close($ch);

    $return_info = array(
        "url" => $url,
        "sent_parameter" => $parameter,
        "http_status" => $retcode,
        "curl_error_no" => $curl_error,
        "web_info" => $result
    );
    return $return_info;
}

//====以上為副程式====

$post_data_array = array( //post_data 欄位資料

    "RespondType" => $RespondType,
    "Version" => "1.0",
    "Amt" => "1",
    "TradeNo" => $TradeNo,
    "MerchantOrderNo" => $MerchantOrderNo,
    "CloseType" => 1,
    "IndexType" => 1,
    "TimeStamp" => time()
);
$post_data_str = http_build_query($post_data_array);
    
```

```

    $post_data = trim(bin2hex(mcrypt_encrypt(MCRYPT_RIJNDAEL_128, $key,
addpadding($post_data_str), MCRYPT_MODE_CBC, $iv))); //加密

    $transaction_data_array = array( //送出欄位
        "MerchantID_" => $merchant_id,
        "PostData_" => $post_data
    );
    $transaction_data_str = http_build_query($transaction_data_array);

    $result = curl_work($url, $transaction_data_str); //背景送出

    if($RespondType=="JSON") {
    echo " <hr>JSON 回應<hr>";

    $value = json_decode($result[web_info],TRUE); //以 json_decode 把內容解為變數陣
列

    echo "回傳狀態:".$value[Status]."<br>";

    echo "回傳訊息:".$value[Message]."<br>";

    echo "商店代號:".$value[Result][MerchantID]."<br>"; //商店代號

    echo "交易金額:".$value[Result][Amt]."<br>";          //取號金額

    echo "智付寶交易序號:".$value[Result][TradeNo]."<br>";    // 智付寶交易序號

    echo "商店訂單編號:".$value[Result][MerchantOrderNo]."<br>";    //商店訂單編號

    }

    if($RespondType=="String") {
    
```



```

        echo " <hr>String 回應<hr>";

        $str=$result['web_info'];
    parse_str(trim($str),$value);

    echo "回傳狀態:". $value[Status]. "<br>";

    echo "回傳訊息:". $value[Message]. "<br>";

    echo "商店代號:". $value[MerchantID]. "<br>"; //商店代號

    echo "交易金額:". $value[Amt]. "<br>";          //取號金額

    echo "智付寶交易序號:". $value[TradeNo]. "<br>";    // 智付寶交易序號

    echo "商店訂單編號:". $value[MerchantOrderNo]. "<br>";    //商店訂單編號

    }
    ?>
    
```

若非使用 PHP 程式語言編寫 Aes 加密機制，可參考下列測試資料與加密後結果：

原始字串：

加密 key=12345678901234567890123456789012

加密 iv=1234567890123456

加密前字串=abcdefghijklmnop

加密後字串：

b91d3ece42c203729b38ae004e96efb9b64c41eeb074cad7ebafa3973181d233

以下提供.net c#版本程式語言編寫 AES 加密語法以供參考：

```
public string EncryptAES256(string source)//加密
{
    string sSecretKey = "12345678901234567890123456789012";
    string iv = "1234567890123456";

    byte[] sourceBytes =
AddPKCS7Padding(Encoding.UTF8.GetBytes(source), 32);
    var aes = new RijndaelManaged();
    aes.Key = Encoding.UTF8.GetBytes(sSecretKey);
    aes.IV = Encoding.UTF8.GetBytes(iv);
    aes.Mode = CipherMode.CBC;
    aes.Padding = PaddingMode.None;

    ICryptoTransform transform = aes.CreateEncryptor();

    return ByteArrayToHex(transform.TransformFinalBlock(sourceBytes, 0,
sourceBytes.Length)).ToLower();
}

public string DecryptAES256(string encryptData)//解密
{
    string sSecretKey = "12345678901234567890123456789012";
    string iv = "1234567890123456";

    var encryptBytes = HexStringToByteArray(encryptData.ToUpper());
    var aes = new RijndaelManaged();
    aes.Key = Encoding.UTF8.GetBytes(sSecretKey);
    aes.IV = Encoding.UTF8.GetBytes(iv);
    aes.Mode = CipherMode.CBC;
    aes.Padding = PaddingMode.None;
    ICryptoTransform transform = aes.CreateDecryptor();

    return
Encoding.UTF8.GetString(RemovePKCS7Padding(transform.TransformFinalBlock(e
ncryptBytes, 0, encryptBytes.Length)));
}
```

```
}

private static byte[] AddPKCS7Padding(byte[] data, int iBlockSize)
{
    int iLength = data.Length;
    byte cPadding = (byte)(iBlockSize - (iLength % iBlockSize));
    var output = new byte[iLength + cPadding];
    Buffer.BlockCopy(data, 0, output, 0, iLength);
    for (var i = iLength; i < output.Length; i++)
        output[i] = (byte)cPadding;
    return output;
}

private static byte[] RemovePKCS7Padding(byte[] data)
{
    int iLength = data[data.Length - 1];
    var output = new byte[data.Length - iLength];
    Buffer.BlockCopy(data, 0, output, 0, output.Length);
    return output;
}

private static string ByteArrayToHex(byte[] barray)
{
    char[] c = new char[barray.Length * 2];
    byte b;
    for (int i = 0; i < barray.Length; ++i)
    {
        b = ((byte)(barray[i] >> 4));
        c[i * 2] = (char)(b > 9 ? b + 0x37 : b + 0x30);
        b = ((byte)(barray[i] & 0xF));
        c[i * 2 + 1] = (char)(b > 9 ? b + 0x37 : b + 0x30);
    }

    return new string(c);
}

private static byte[] HexStringToByteArray(string hexString)
{

```

```
int hexStringLength = hexString.Length;
byte[] b = new byte[hexStringLength / 2];
for (int i = 0; i < hexStringLength; i += 2)
{
    int topChar = (hexString[i] > 0x40 ? hexString[i] - 0x37 : hexString[i]
- 0x30) << 4;
    int bottomChar = hexString[i + 1] > 0x40 ? hexString[i + 1] - 0x37 :
hexString[i + 1] - 0x30;
    b[i / 2] = Convert.ToByte(topChar + bottomChar);
}
return b;
}
```