

藍新金流服務平台

信用卡-取消授權

技術串接手冊

標準版

文件版本號：V 1.0.2

(文件為藍新科技股份有限公司版權所有)

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	2018/12/24
2	V 1.0.1	1. 【第五章 信用卡取消授權參數設定】移除 〈NotifyURL 批次授權結果通知網址〉參數 2. 移除【第六章 幕後通知回傳參數說明】章節 3. 【第七章 錯誤代碼】新增錯誤代碼	2020/02/20
3	V 1.0.2	1. 【第七章】新增錯誤代碼 TRA11002	2020/06/22

目 錄

一、串接目的.....	3
二、作業流程圖.....	4
三、測試環境串接與作業流程.....	5
四、正式環境串接與作業流程.....	6
五、信用卡取消授權參數設定.....	7
六、取消授權完成後系統回應訊息.....	8
七、錯誤代碼.....	10
附件一 PostData_加密方法	12
附件二 CheckCode 產生規則	17

一、串接目的

商店發動請款前，針對單筆授權完成的交易發送取消授權指令至藍新金流，藍新金流接收指令後進行取消授權作業，並以 Web Services 方式回應商店，以利商店後續作業。

藍新金流提供之金流工具如下：

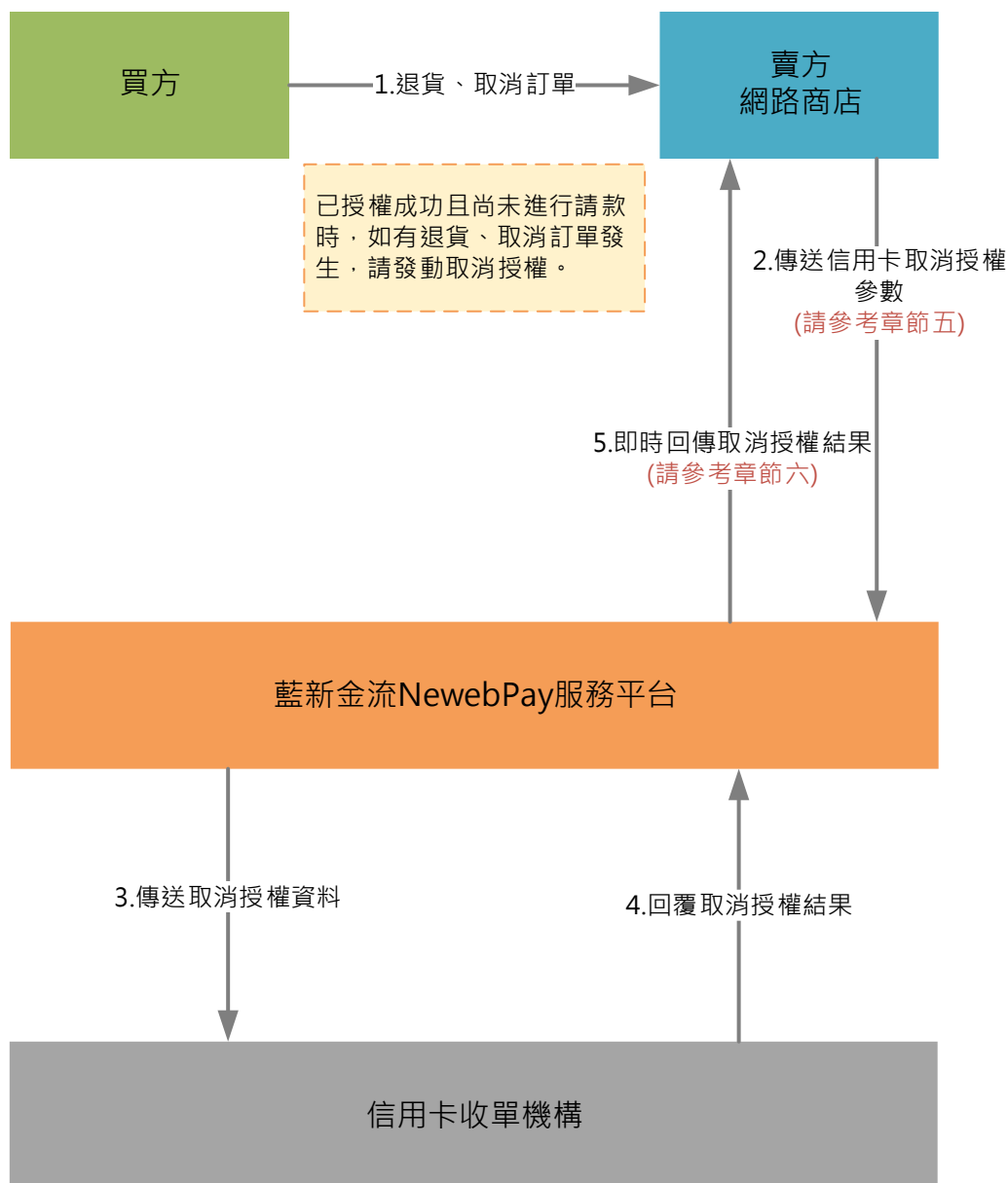
藍新金流支付工具總類	支付工具支援模式
信用卡(CREDIT)	代收模式、闖道模式

備註：

■ 信用卡(CREDIT)

1. 提供一次付清方式 (包含：國外卡、動態貨幣轉換 DCC、Google Pay、Samsung Pay)、分期付款、紅利折抵方式。
2. 支援 Visa、Master、JCB、美國運通卡、銀聯等卡別
3. 動態貨幣轉換 DCC 僅支援台新銀行一次付清之代收商店(支援卡別 Visa、Master)。
4. 提供國民旅遊卡資料主動報送機制，可協助與收單機構直接簽約的國民旅遊卡收單特約商店，省去人工逐筆登錄國民旅遊卡資料的繁瑣作業。
5. 提供信用卡快速結帳功能，可提供付款人選擇是否讓藍新金流記憶信用卡卡號，作為下次支付時使用。
6. 分期付款、紅利扣抵之請款與退款為全額請退款，無法部分請退款。

二、作業流程圖



1. 回覆取消授權結果，依據信用卡收單金融機構規定不同，分為 [即時處理] 與 [批次處理] 兩種模式。
2. 商店可至會員中心查看取消授權結果。

三、測試環境串接與作業流程

(一) 測試串接網址：<https://ccore.newwebpay.com/API/CreditCard/Cancel>

1. 請於藍新金流測試平台註冊會員並建立測試商店。
 - (1) 請至藍新金流測試平台 <https://cwww.newwebpay.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) 請登入藍新金流測試平台，於【會員中心】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. 測試網站僅接受測試卡號：4000-2211-1111-1111 交易成功。 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】：您於藍新金流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。

四、正式環境串接與作業流程

(一) 正式串接網址：<https://core.newwebpay.com/API/CreditCard/Cancel>

1. 請於藍新金流平台註冊會員並建立商店。

(1) 請至藍新金流平台 <https://www.newwebpay.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) 請登入藍新金流平台，於【會員中心】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. 正式開始使用藍新金流平台服務。

(二) 資料交換方式：

(1) 商店以幕後「HTTP POST」方式傳送授權交易至藍新金流平台進行取消授權。

(2) 傳送方式須以後端 Form Post 進行。

(3) 藍新金流將以 Web Services 方式回應。

(4) 編碼格式為 UTF-8。

(三) 信用卡取消授權執行條件：

(1) 取消授權交易須為授權成功交易。

(2) 取消授權必須在發動請款之前執行(即自交易授權日起算 21 個日曆日晚上九點前)。

(3) 不可重覆取消。

(4) 完成取消授權的交易，將會於信用卡取消授權後，返還藍新金流商店及持卡人信用卡授權額度。

五、信用卡取消授權參數設定

(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(10)	藍新金流商店代號。
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	取消授權金額	V	Int(10)	純數字不含符號，需與授權金額相同
MerchantOrderNo	商店訂單編號	+	Varchar(30)	與藍新金流交易序號二擇一填入
TradeNo	藍新金流交易序號	+	Varchar(17)	與商店訂單編號二擇一填入
IndexType	單號類別	V	Int(1)	只限定填數字 1 或 2，1 表示使用商店 訂單編號，2 表示使用藍新金流交易單 號
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

註：必填欄位中，符號”V”為必填；”+ ”為選填，”商店訂單編號”與”藍新金流交易序號”兩欄位必定要帶入一項，且於 IndexType 指定。

六、取消授權完成後系統回應訊息

(一) 藍新金流平台接收到取消授權資訊後，回應結果並顯示於系統頁面。

(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 方式來接收以下 JSON 格式的參數資料，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
Status	回傳狀態	Varchar(10)	1.若取消授權程序成功，則回傳 SUCCESS。 2.若此筆交易取消授權需由金融機構批次處理，則回傳代碼：TRA20001。 3.若失敗則回傳錯誤代碼，錯誤代碼請參考 “七、錯誤代碼” 。
Message	回傳訊息	Varchar(30)	文字，敘述此次取消授權狀態
Result	回傳資料	Array	內容格式為陣列

上述 Result 欄位內含下列參數：

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
TradeNo	藍新金流交易序號	Varchar(20)	藍新金流在此筆交易成立時所產生的序號
Amt	交易金額	Int(15)	本次交易取消授權金額
MerchantOrderNo	商店訂單編號	Varchar(30)	此取消授權動作所執行的商店訂單編號
CheckCode	檢核碼	Hash	用來檢查此次資料回傳的合法性，商戶串接時可以比對此欄位資料來檢核是否為藍新金流平台所回傳，檢核方法請參考 附件二

範例程式：

```
{
  "Status": "SUCCESS",
  "Message": "\u653e\u68c4\u6388\u6b0a\u6210\u529f",
  "Result": {
    "MerchantID": "3430112",
    "Amt": 30,
    "MerchantOrderNo": "1453973102",
    "TradeNo": "16012817252770139",
    "CheckCode": "2214FFC8D10B7C207E0844AF7F9A6C8B396649A643953A51D9F73086B55678A0"
  }
}
```

(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.若取消授權程序成功，則回傳 SUCCESS。 2.若此筆交易取消授權需由金融機構批次處理，則回傳代碼：TRA20001 3.若失敗則回傳錯誤代碼，錯誤代碼請參考 “七、錯誤代碼” 。
Message	回傳訊息	Varchar(30)	文字，敘述此次取消授權狀態
MerchantID	商店代號	Varchar(15)	商店代號
Amt	交易金額	Int(15)	本次交易取消授權金額
TradeNo	藍新金流交易序號	Varchar(20)	藍新金流在此筆交易成立時所產生的序號
MerchantOrderNo	商店訂單編號	Varchar(30)	此取消授權動作所執行的商店訂單編號
CheckCode	檢核碼	Hash	用來檢查此次資料回傳的合法性，商戶串接時可以比對此欄位資料來檢核是否為藍新金流平台所回傳，檢核方法請參考 附件二

範例程式：

```
Status=SUCCESS&Message=%E6%94%BE%E6%A3%84%E6%8E%88%E6%AC%8A%E6%88%90%E5%8A%9F&MerchantID=3430112&Amt=30&MerchantOrderNo=1453973047&TradeNo=16012817244115269&CheckCode=8F70A9E59902B12DA13A3635C7CB2C45C130563B747FA431D66075E57A9A28A9
```

七、錯誤代碼

商店傳送取消授權資料時，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
ACC10005	會員已被暫時停權/永久停權， 請洽藍新金流客服中心查詢	
MEM40008	必填欄位資料不可空白	會提示空白欄位
MEM40012	必填欄位資料傳遞錯誤	PostData_空白
MEM40013	必填欄位資料不齊全	會提示空白欄位
NOR1001	連線異常	
TRA10001	商店代號錯誤	商店資料取得失敗
TRA10003	金額填入非數字資訊	
TRA10008	資料加密錯誤	PostData_解密失敗
TRA10009	商店代號空白	商店代號不可空白
TRA10012	商店代號停用	
TRA10013	商店信用卡資格停用	
TRA10015	網路連線失敗	
TRA10021	查無該筆交易或該筆交易不為信用卡交易	
TRA10032	單號類別錯誤	
TRA10033	商店訂單編號空白或藍新金流交易序號空白	
TRA10036	RespondType 欄位資料格式錯誤	
TRA10037	商家自訂單號錯誤	
TRA10038	藍新金流交易序號格式錯誤	
TRA10047	交易非授權成功狀態	
TRA10048	該交易正在請款狀態	
TRA10049	該交易正在退款狀態	
TRA10050	金額不符	
TRA10051	取消授權失敗	

TRA11002	因失敗過多您的商店暫時無法使用本功能， 請待 4 小時解鎖	
TRA20001	金融機構取消授權批次處理中	此交易取消授權，需由金融機構 批次處理
TRA40014	TimeStamp 欄位錯誤	

附件一 PostData_加密方法

A. PostData_加密方法(以 PHP 為例)：

使用此串接規格時，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，再以幕後方式送至信用卡取消授權閘道。

所有資料與欄位使用 '=' 符號組合，再用 '&' 符號串起字串，例如：

ResponseType=Json&Version=1.0(以此類推)

以下提供 PHP 版本程式語言範例以供參考：

```
<?php
$post_data_str = 'ResponseType=Json&Version=1.0';
    $post_data = trim(bin2hex(mcrypt_encrypt(MCRYPT_RIJNDAEL_128,
$key, addpadding($post_data_str), MCRYPT_MODE_CBC, $iv))); //加密

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

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 key=1234567890123456

加密前字串=abcdefghijklmnopqrstuvwxyABCDEF

加密後字串：

**b91d3ece42c203729b38ae004e96efb90109ee25f7861b6bb33891be88d9a7994
84f0d3ccee9a094e9fad6d51db716ff2df7a5137639aaf94fba4f309e2af173**

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
    (encryptBytes, 0, encryptBytes.Length)));
}
```

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

private static byte[] RemovePKCS7Padding(byte[] data)
{
intiLength = 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 (inti = 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)
{
    inthexStringLength = hexString.Length;
    byte[] b = new byte[hexStringLength / 2];
```



```
        for (inti = 0; i<hexStringLength; i += 2)
        {
            inttopChar = (hexString[i] > 0x40 ? hexString[i] - 0x37 : hexString[i] - 0x30) << 4;
            intbottomChar = hexString[i + 1] > 0x40 ? hexString[i + 1] - 0x37 : hexString[i + 1] -
            0x30;

            b[i / 2] = Convert.ToByte(topChar + bottomChar);
        }
        return b;
    }
```

附件二 CheckCode 產生規則

1. 排序欄位字串並用&符號串聯起來

將回傳的資料其中的四個欄位，分別是 Amt(金額)、MerchantID(商店代號)、MerchantOrderNo(商店訂單編號)、TradeNo(藍新金流交易序號)，且參數需照英文字母 A~Z 排序，若第一字母相同比較第二字母，以此類推。

2. 將串聯後的字串前後加上商店專屬加密 HashIV 值與商店專屬加密 HashKey 值，

3. 將串聯後的字串用 SHA256 壓碼後轉大寫。

範例程式：

(1) 排列參數欄位並串聯

```
Amt=100&MerchantID=1422967&MerchantOrderNo=840f022&TradeNo=14061313541640927
```

(2) 前後加上商店專屬的 Hash Key 及 Hash IV

```
$check_code_str =  
HashIV=1234567&Amt=100&MerchantID=1422967&MerchantOrderNo=840f022&TradeNo=14061313541640927&HashKey=abcdefg
```

(3) 使用 SHA256 壓碼過後並轉大寫。字串結果如下：

```
strtoupper(hash("sha256", $check_code_str))
```

以下提供 PHP 版本程式語言範例以供參考：

```
$check_code = array(  
    "MerchantID" => '1422967',  
    "Amt" => '100',  
    "MerchantOrderNo" => '840f022',  
    "TradeNo" => '14061313541640927',  
);  
ksort($check_code);  
$check_str = http_build_query($check_code);  
$CheckCode_str = "HashIV=1234567&$check_str&HashKey=abcdefg";  
$CheckCode = strtoupper(hash("sha256", $CheckCode_str));
```

**\$CheckCode=62C687AF6409E46E79769FAF54F54FE7E75AAE50BAF0767752A5C3
37670B8EDB**