



綠界科技股份有限公司

Green World FinTech Service Co., Ltd.

站內付 2.0

新型閘道技術文件_Web 版

V 1.0.3

文件編號 gw_p702

2023-09-08

目錄

1.	簡介.....	1
2.	系統介接測試相關資訊.....	2
3.	一般交易流程說明.....	3
(1)	信用卡作業流程(含 3D 驗證).....	3
(2)	信用卡作業流程(不含 3D 驗證).....	5
(3)	非信用卡作業流程.....	7
(4)	銀聯卡作業流程.....	9
(5)	Apple Pay 作業流程.....	11
4.	取得廠商驗證碼(一般交易功能/Server)	13
(1)	交易付款.....	13
(2)	刪除記憶信用卡資料.....	21
5.	一般交易嵌入式畫面 SDK(JS).....	24
6.	建立一般交易(Server).....	29
7.	定期定額查詢(Server).....	36
8.	信用卡定期定額訂單作業(Server).....	41
9.	查詢 ATM/CVS/BARCODE 取號結果(Server)	44
10.	綁定信用卡流程說明.....	48
11.	綁定信用卡使用情境說明.....	50

12.	取得廠商驗證碼(綁定信用卡功能/Server)	53
13.	綁定信用卡嵌入式畫面 SDK(JS).....	58
14.	建立綁定信用卡訂單(Server).....	61
15.	查詢會員綁定信用卡資料(Server).....	64
16.	刪除綁定信用卡(Server).....	68
17.	幕後交易授權(Server).....	71
18.	付款結果通知.....	76
19.	查詢訂單(Server).....	88
20.	查詢信用卡單筆明細記錄(Server).....	93
21.	信用卡請退款功能(Server).....	97
附錄 1.	參數加密方式說明.....	102
附錄 2.	交易訊息代碼一覽表.....	103
附錄 3.	CSS 參數定義列表.....	104
附錄 4.	銀行代碼表.....	105

Version History :

Version	Date	Content
V1.0.0	2022/07/04	Create
V1.0.1	2023/01/03	調整交易金額參數說明
V1.0.2	2023/08/31	調整文件標題
V1.0.3	2023/09/08	特約會員調整文字為特約賣家

1. 簡介

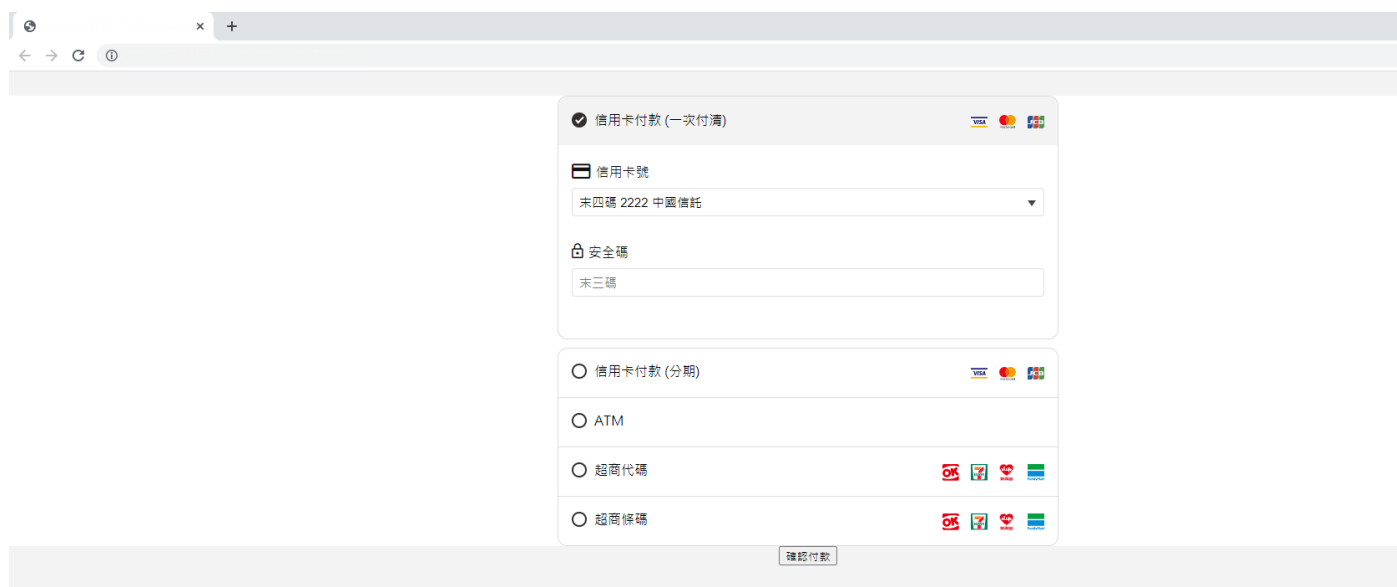
綠界站內付 2.0 服務對於有收款需求的特店，提供完整的交易介接技術與多元的付款方式，並且提供 Web 與 APP 兩種串接服務；此份規格主要說明 Web JS 串接，功能包含信用卡、ATM、超商代碼與超商條碼可供選擇。特店可將站內付 2.0 產生的付款畫面嵌入至自己的網頁，並可自行修改付款畫面樣式，以達到自行設計付款畫面樣式的需求。

如欲使用站內付 2.0，您必須成為綠界特約賣家並且申請開通。請前往以下網址並填寫資料，我們將派專人與您聯繫。

<https://member.ecpay.com.tw/MemberReg/MerchantRegister>

※注意事項:

此份串接規格為 Web 版；如有 APP 串接需求，請洽參考站內付 2.0 介接技術文件(閘道商專用)_App。



The screenshot displays a web browser window with a payment form. The form is titled '信用卡付款 (一次付清)' (Credit Card Payment (One-time Payment)). It includes a dropdown menu for '信用卡號' (Credit Card Number) showing '末四碼 2222 中國信託' (Last four digits 2222, China Trust). Below this is a '安全碼' (Security Code) field with '末三碼' (Last three digits) as a placeholder. To the right of the form are logos for VISA, Mastercard, and UnionPay. Below the main form are four radio button options: '信用卡付款 (分期)' (Credit Card Payment (Installment)), 'ATM', '超商代碼' (Convenience Store Code), and '超商條碼' (Convenience Store Barcode). Each option has corresponding logos to its right. At the bottom right of the form is a '確認付款' (Confirm Payment) button.

站內付 2.0 Web JS 示意圖

2. 系統介接測試相關資訊

請使用以下資訊介接本公司金流服務。此為測試環境，測試沒問題後，要連正式環境時，請將以下資訊更換成正式環境的相關資訊。

欄位說明	欄位內容	
特店編號(MerchantID)	3002607	--
閘道商特店編號 (GatewayMerchantID)	--	3085779
登入廠商後台帳號	stagetest3	gatewaytest02
登入廠商後台密碼	test1234	
統一編號	00000000	
廠商後台測試環境	https://vendor-stage.ecpay.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。	
all in one 介接的 HashKey	pwFHCqoQZGmho4w6	y6869NBszTuvhSRx
all in one 介接的 HashIV	EkRm7iFT261dpevs	BMm7FmX91dE8rpdw
信用卡測試卡號	4311-9522-2222-2222(注意事項:只有此組卡號可測試交易成功流程)	
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。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。	

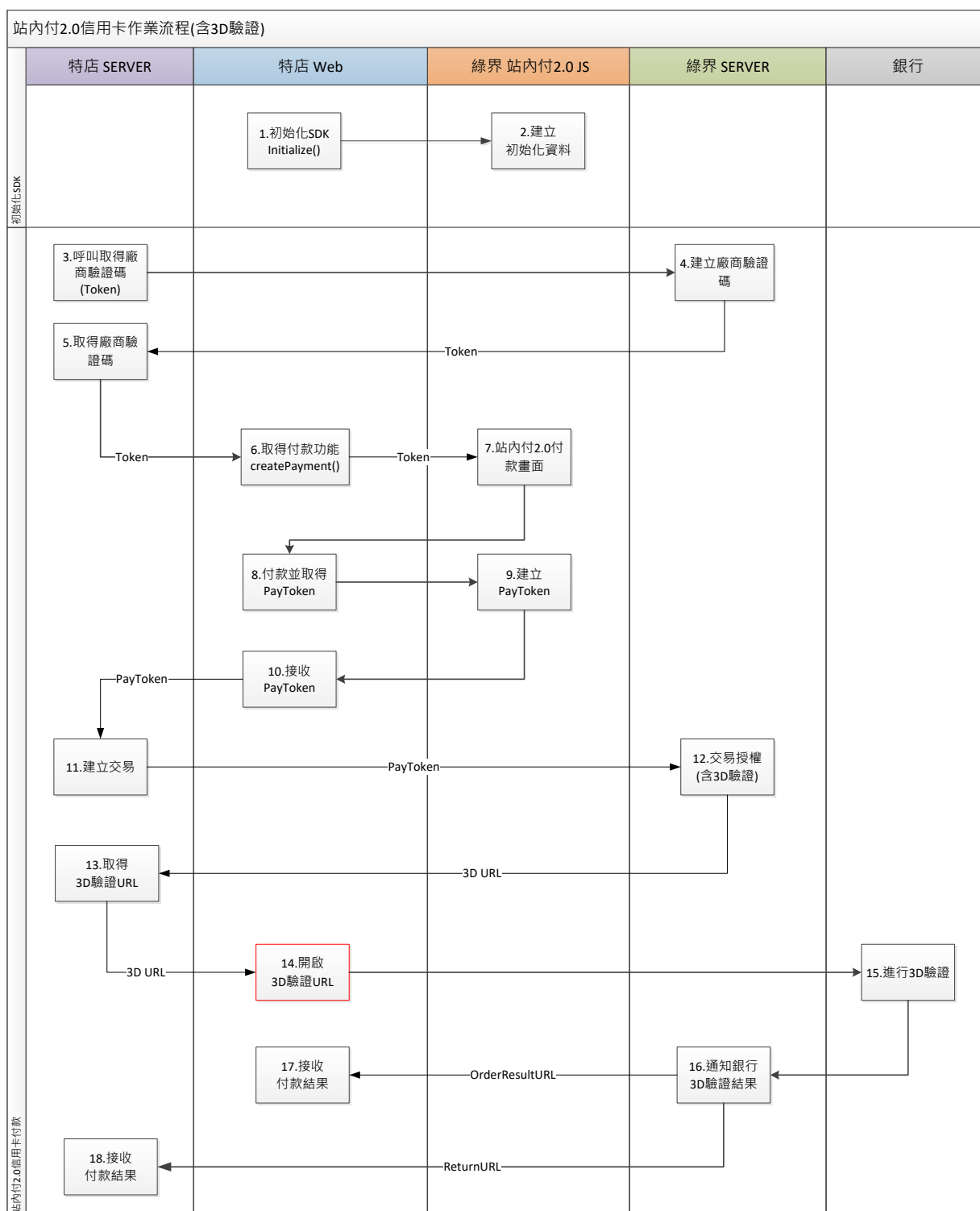
※注意事項：

1. 接收傳送綠界 API 通知時，請開發人員確認下面事項，以利正常收到綠界發送的各项通知：
2. 請確認 Server 是否有開通防火牆，以避免付款通知被防火牆阻擋。
3. 呼叫 API 連接 port 只提供 https (443 port)連線方式，並請使用 DNS(Domain Name System)進行介接。
4. 請確認各項交易參數傳送時是使用 Http POST 方式傳送至綠界 API。
5. 請確認 Server URL 連接 port 為 http 80 port 與 https 443 port。
6. 測試環境信用卡付款方式未提供海外卡及銀聯卡服務。
7. 傳送參數值的內容請用 UTF8 格式，且不支援中英文語系之外的文字內容。
8. 請先申請關閉 OTP 服務及申請開啟信用卡 3D 驗證服務才能完成串接交易。
9. 為保障消費者權益與網路交易安全，串接服務僅支援 TLS 1.1 以上之加密通訊協定。
10. 綠界主機 IP 不固定，如廠商防火牆需開通綠界 IP，請以 FQDN 方式設定以下 domain:
連入:postgate.ecpay.com.tw、postgate-stage.ecpay.com.tw
連出:payment.ecpay.com.tw、payment-stage.ecpay.com.tw
11. Apple Pay 功能不提供測試環境使用，請於正式環境產生憑證進行測試。

3. 一般交易流程說明

一般交易的使用，適用於信用卡一次付清、分期付款、定期定額或非信用卡等多種付款方式。
以下將分別說明不同的付款方式所使用的流程：

(1) 信用卡作業流程(含 3D 驗證)

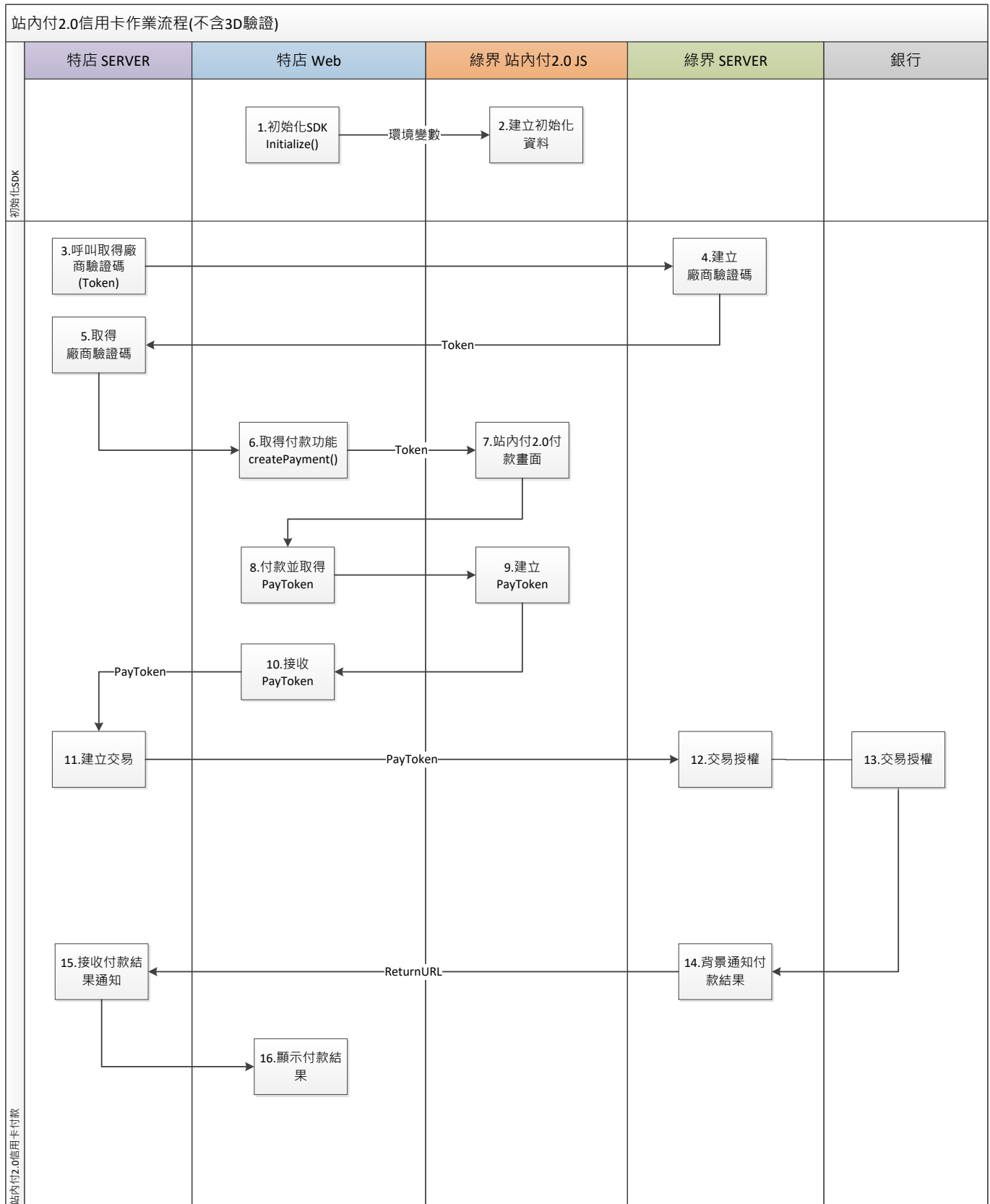


處理角色	流程名稱	處理說明
特店 Web	1.初始化 SDK	開始使用 SDK 需先初始化，並帶入環境變數
綠界站內付 2.0 JS	2.建立 SDK 初始化資料	SDK 建立初始化資料
特店 Server	3.呼叫取得 廠商驗證碼	特店確定訂單內容後，傳送訂單資料到綠界並取得廠商驗證碼 (Token)。API 詳細說明請參考 取得廠商驗證碼(一般交易功能) 章節
綠界 Server	4.建立廠商驗證碼	綠界 Server 接收到訂單資料後，產生一組廠商驗證碼(Token)
特店 Server	5.取得廠商驗證碼	接收綠界回傳的廠商驗證碼(Token)
特店 Web	6.取得付款功能	呼叫取得付款功能
綠界站內付 2.0 JS	7.站內付 2.0 付款畫面	將取得的 Token 帶入 JS SDK，產生付款畫面。詳細說明請參考 一般交易嵌入式畫面 SDK 章節。
特店 Web	8.付款並取得 PayToken	呼叫 一般交易嵌入式畫面 SDK 章節提供的 createPayment()取得 PayToken
綠界站內付 2.0 JS	9.建立 PayToken	建立 PayToken 資料
特店 Web	10.接收 PayToken	接收 PayToken 並將 PayToken 傳至特店 Server
特店 Server	11. 建立交易	呼叫建立一般交易 API
綠界 Server	12.交易授權(含 3D 驗證)	將交易資料送至綠界 Server 建立訂單並將信用卡資料送至銀行進行授權與 3D 驗證
特店 Server	13.取得 3D 驗證 URL	取得 3D 驗證 URL 並將 URL 傳送至特店 Web
特店 Web	14.開啟 3D 驗證 URL	開啟 3D 驗證 URL
銀行	15.進行 3D 驗證	銀行進行 3D 驗證
綠界 Server	16.通知銀行 3D 驗證結果	接收 3D 驗證結果
特店 Web	17.接收付款結果	透過 OrderResultURL 接收付款結果
特店 Server	18.接收付款結果	透過 ReturnURL 背景接收付款結果通知

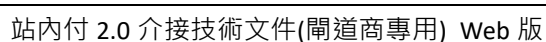
※注意事項：

步驟 14：當消費者選擇信用卡付款時，特店必須利用取得的 3D 驗證 URL 實作 3D 驗證導轉頁面，提供消費者信用卡 3D 驗證作業，進行信用卡授權。

(2) 信用卡作業流程(不含 3D 驗證)

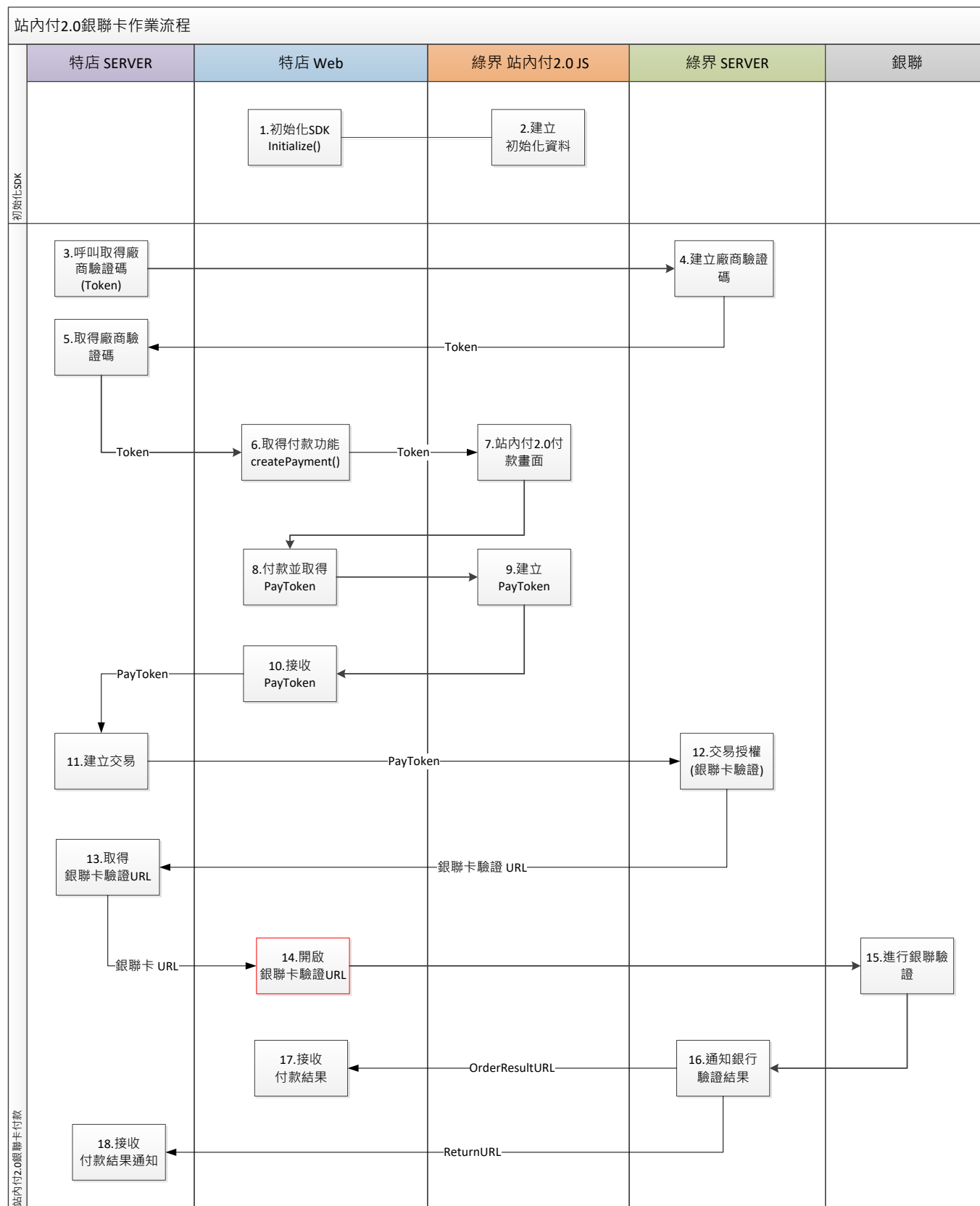


處理角色	流程名稱	處理說明
特店 Web	1.初始化 SDK	開始使用 SDK 需先初始化，並帶入環境變數
綠界站內付 2.0 JS	2.建立 SDK 初始化資料	SDK 建立初始化資料
特店 Server	3.呼叫取得 廠商驗證碼	特店確定訂單內容後，傳送訂單資料到綠界並取得廠商驗證碼。API 詳細說明請參考 取得廠商驗證碼(一般交易功能) 章節
特店 Server	4.建立廠商驗證碼	綠界 Server 接收到訂單資料後，產生一組廠商驗證碼
特店 Server	5.取得廠商驗證碼	接收綠界回傳的廠商驗證碼
特店 Web	6.取得付款功能	呼叫取得付款功能
綠界站內付 2.0 JS	7. 站內付 2.0 付款畫面	將取得的 Token 帶入 JS SDK，產生付款畫面。詳細說明請參考 一般交易嵌入式畫面 SDK 章節。
特店 Web	8.付款並取得 PayToken	呼叫站內付 2.0 JS 提供的 createPayment()取得 PayToken
綠界站內付 2.0 JS	9.建立 PayToken	建立 PayToken 資料
特店 Web	10.接收 PayToken	接收 PayToken 並將 PayToken 傳至特店 Server
特店 Server	11. 建立交易	呼叫建立交易 API
綠界 Server	12.交易授權(不含 3D 驗證)	將交易資料送至綠界 Server 建立訂單並將信用卡資料送至銀行進行授權
銀行	13.交易授權	進行信用卡交易授權作業，並將授權結果回應給綠界
綠界 Server	14.背景通知付款結果	透過 ReturnURL 背景通知付款結果
特店 Server	15.接收付款結果通知	透過 ReturnURL 背景接收付款結果通知
特店 Web	16.顯示付款結果	顯示付款結果



處理角色	流程名稱	處理說明
特店 Web	1.初始化 SDK	開始使用 SDK 需先初始化，並帶入環境變數
綠界站內付 2.0 JS	2.建立 SDK 初始化資料	SDK 建立初始化資料
特店 Server	3.呼叫取得 廠商驗證碼	特店確定訂單內容後，傳送訂單資料到綠界並取得廠商驗證碼。API 詳細說明請參考 取得廠商驗證碼(一般交易功能) 章節
特店 Server	4.建立廠商驗證碼	綠界 Server 接收到訂單資料後，產生一組廠商驗證碼
特店 Server	5.取得廠商驗證碼	接收綠界回傳的廠商驗證碼
特店 Web	6.取得付款功能	呼叫取得付款功能
綠界站內付 2.0 JS	7.站內付 2.0 付款畫面	將取得的 Token 帶入 JS SDK，產生付款畫面。詳細說明請參考 一般交易嵌入式畫面 SDK 章節。
特店 Web	8.付款並取得 PayToken	呼叫嵌入式 JS 提供的 createPayment()取得 PayToken
綠界站內付 2.0 JS	9.建立 PayToken	建立 PayToken 資料
特店 Web	10.接收 PayToken	接收 PayToken 並將 PayToken 傳至特店 Server
特店 Server	11. 建立交易	呼叫建立交易 API
綠界 Server	12.建立交易	將交易資料送至綠界 Server 建立訂單並取得取號結果
特店 Server	13.取得取號結果	取得綠界回傳取號結果資料
特店 Web	14.顯示繳費代碼	顯示繳費代碼提供消費者進行繳費
銀行/超商	15.確認繳費	當消費者完成付款，通知綠界完成繳費
綠界 Server	16.背景通知付款結果	透過 ReturnURL 背景通知特店付款結果
特店 Server	17.接收付款結果通知	透過 ReturnURL 背景接收付款結果通知

(4) 銀聯卡作業流程

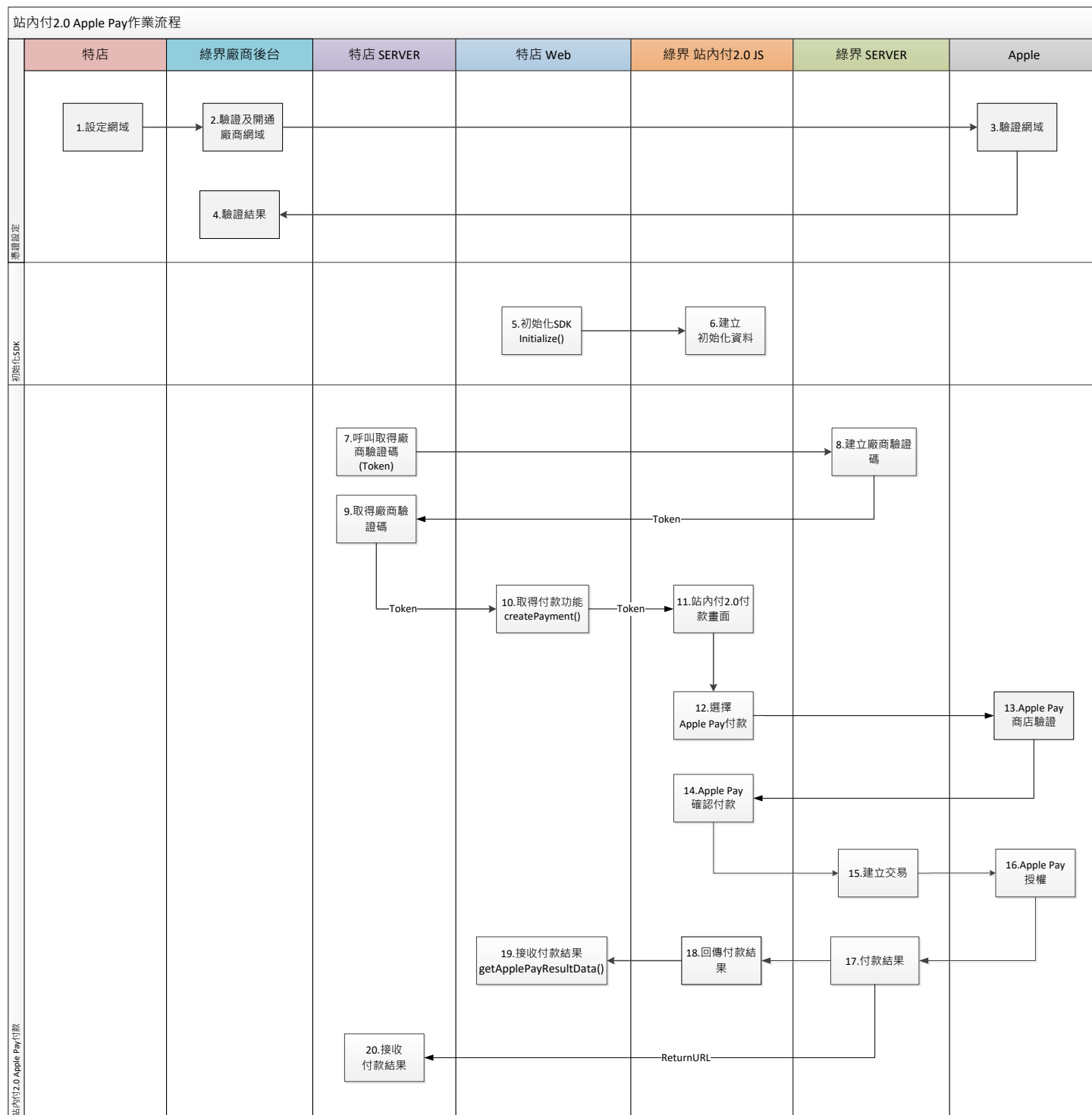


處理角色	流程名稱	處理說明
特店 Web	1.初始化 SDK	開始使用 SDK 需先初始化，並帶入環境變數
綠界站內付 2.0 JS	2.建立 SDK 初始化資料	SDK 建立初始化資料
特店 Server	3.呼叫取得 廠商驗證碼	特店確定訂單內容後，傳送訂單資料到綠界並取得廠商驗證碼。API 詳細說明請參考 取得廠商驗證碼(一般交易功能) 章節
綠界 Server	4.建立廠商驗證碼	綠界 Server 接收到訂單資料後，產生一組廠商驗證碼(Token)
特店 Server	5.取得廠商驗證碼	接收綠界回傳的廠商驗證碼(Token)
特店 Web	6.取得付款功能	將取得的 Token 帶入 JS SDK，產生付款畫面。詳細說明請參考 一般交易嵌入式畫面 SDK 章節。
綠界站內付 2.0 JS	7.站內付 2.0 付款畫面	將取得的 Token 帶入 JS SDK，產生付款畫面。詳細說明請參考 一般交易嵌入式畫面 SDK 章節。
特店 Web	8.付款並取得 PayToken	呼叫站內付 2.0 JS 提供的 createPayment()取得 PayToken
綠界站內付 2.0 JS	9.建立 PayToken	建立 PayToken 資料
特店 Web	10.接收 PayToken	接收 PayToken 並將 PayToken 傳至特店 Server
特店 Server	11. 建立交易	呼叫建立交易 API
綠界 Server	12.交易授權(銀聯驗證)	將交易資料送至綠界 Server 建立訂單並將信用卡資料送至銀行進行授權
特店 Server	13.取得銀聯驗證 URL	取得銀聯驗證 URL 並將 URL 傳送至特店 Web
特店 Web	14.開啟銀聯驗證 URL	開啟銀聯驗證 URL
銀聯	15.進行銀聯驗證	銀行進行驗證
綠界 Server	16.通知銀行銀聯驗證結果	接收銀聯驗證結果
特店 Web	17.接收付款結果	透過 OrderResultURL 接收付款結果
特店 Server	18.接收付款結果通知	透過 ReturnURL 背景接收付款結果通知

※注意事項：

步驟 14：當消費者選擇銀聯卡付款時，特店必須利用取得的銀聯驗證 URL 實作驗證導轉頁面，提供消費者銀聯卡驗證作業，進行信用卡授權。

(5) Apple Pay 作業流程



處理角色	流程名稱	處理說明
特店	1.設定網域	第三方支付版 Apple Pay 無需申請 Apple Developer 帳號。 特店在串接 Apple Pay 功能之前，必須先至綠界廠商後台>系統開發管裡>Apple Pay 介接設定，設定 Apple Pay 網域。詳細操作說明可參考 廠商後台操作文件的 Apple Pay 介接設定與 Apple Pay 憑證設定說明 。
綠界廠商後台	2.驗證及開通廠商網域	綠界廠商後台固定在每日的 15:30 分進行驗證，特店必須提早將下載檔案放置於對應路徑。
Apple	3.驗證網域	綠界將驗證資料送至 Apple
綠界廠商後台	4.驗證結果	驗證結果預計於每日的 16:00 前更新
特店 Web	5.初始化 SDK	開始使用 SDK 需先初始化，並帶入環境變數
綠界站內付 2.0 JS	6.建立 SDK 初始化資料	SDK 建立初始化資料
特店 Server	7.呼叫取得 廠商驗證碼	特店確定訂單內容後，傳送訂單資料到綠界並取得廠商驗證碼。API 詳細說明請參考 取得廠商驗證碼(一般交易功能) 章節
綠界 Server	8.建立廠商驗證碼	綠界 Server 接收到訂單資料後，產生一組廠商驗證碼(Token)
特店 Server	9.取得廠商驗證碼	接收綠界回傳的廠商驗證碼(Token)
特店 Web	10.取得付款功能	將取得的 Token 帶入 JS SDK，產生付款畫面。詳細說明請參考 一般交易嵌入式畫面 SDK 章節。
綠界站內付 2.0 JS	11.站內付 2.0 付款畫面	將取得的 Token 帶入 JS SDK，產生付款畫面。詳細說明請參考 一般交易嵌入式畫面 SDK 章節。
綠界站內付 2.0 JS	12.選擇 Apple Pay 付款	使用者於付款畫面選擇 Apple Pay 付款
Apple	13. Apple Pay 商店驗證	將商店資料送至 Apple 進行商店驗證
綠界站內付 2.0 JS	14. Apple Pay 確認付款	付款畫面顯示 Apple Pay 付款按鈕，並按下確認付款
綠界 Server	15. 建立交易	建立交易訂單
Apple	16. Apple Pay 授權	綠界將交易訂單送至 Apple 進行授權
綠界 Server	17.付款結果	收到 Apple 回傳的授權結果
綠界站內付 2.0 JS	18.回傳付款結果	收到付款結果後，將結果回傳至 getApplePayResultData()
特店 Web	19.接收付款結果	特店可透過自行實做 getApplePayResultData() ，取得付款結果
特店 Server	20.接收付款結果	透過 ReturnURL 接收付款結果

4. 取得廠商驗證碼(一般交易功能/Server)

(1) 交易付款

- 應用場景：在使用綠界站內付 2.0 金流服務之前，特店 Server 必須先向綠界 Server 取得一組廠商驗證碼(Token)。特店 Server 得到驗證碼後，必須將驗證碼傳給特店 Web 做產生站內付 2.0 金流畫面功能之用。
- 介接路徑
 - 正式環境：https://ecpg.ecpay.com.tw/Merchant/GetTokenbyTrade
 - 測試環境：https://ecpg-stage.ecpay.com.tw/Merchant/GetTokenbyTrade
- 特店傳入參數說明：(JSON 格式，名稱若有紅色星號*為必填欄位)
 - Content Type : application/json
 - HTTP Method : POST

參數	參數名稱	型態	說明
*MerchantID	特店編號	String(10)	特店編號 若為平台商時，參數請帶平台商所綁的特店編號
*RqHeader	傳輸資料	JSON	
*Timestamp	傳輸時間	Number	請將傳輸時間轉換為時間戳(GMT+8)，綠界會利用此參數將當下的時間轉為 Unix TimeStamp 來驗證此次介接的時間區間。 注意事項： 若時間戳跟綠界伺服器接收到時間超過 10 分鐘時，交易會失敗無法進行
*Revision	串接文件版號	String(10)	請參考文件封面的文件版號。Ex: 1.0.0
*Data	加密資料	String	加密過 JSON 格式的資料。 加密方法說明

範例:

```
{
  "MerchantID": "3002607",
  "RqHeader": {
    "Timestamp": 1234567890,
    "Revision": "1.0.0"
  },
  "Data": "..."
}
```

● Data 參數說明：請在加密前對參數值進行 URLEncode

參數	參數名稱	型態	說明
PlatformID	特約合作平台商代號	String(10)	為專案合作的平台商使用。一般特店或平台商本身介接，則參數請帶空值。若為專案合作平台商的特店使用時，則參數請帶平台商所綁的特店編號 [MerchantID]。
*GatewayMerchantID	閘道商特店編號	String(10)	實際建立訂單的金流模式，會依照綠界廠商後台的閘道介接設定內容為主。
*RememberCard	是否使用記憶卡號功能	Int	此功能會將消費者的信用卡記憶在 GatewayMerchantID 上 0:否 1:是
*PaymentUIType	畫面的呈現方式	Int	此欄位為指定付款頁面呈現方式，綠界會依據傳送的内容顯示畫面。 0: 信用卡定期定額 1: 國民旅遊卡 2: 付款選擇清單頁
ChoosePaymentList	欲使用的付款方式	String(30)	PaymentUIType 如選擇 2，則必填 0: 全部付款方式 1: 信用卡一次付清 2: 信用卡分期付款 3: ATM 4: 超商代碼 5: 超商條碼 6: 銀聯卡 7: Apple Pay (第三方支付版 Apple Pay 無需申請 Apple Developer 帳號) 可多選 ex: 1,2,3
*OrderInfo	訂單資訊	JSON	
	*MerchantTradeDate	String(20)	yyyy/MM/dd HH:mm:ss
	*MerchantTradeNo	String(20)	均為唯一值，不可重複使用。英數字大小寫混和 如何避免訂單編號重複請參考 FAQ 。
	*TotalAmount	Int	請帶整數，不可有小數點。 僅限新台幣，且不可超過 20 萬。其餘各付款金額的限制，請參考 會員服務介紹
	*ReturnURL	String(200)	當消費者付款完成後，綠界會將付款結果參數以幕後(Server POST)回傳到該網址。詳細說明請參考付款結果通知。 1. 請勿設定與 Client 端接收付款結果網址 OrderResultURL 相同位置，避免程式判斷錯誤。

				2. 請在收到 Server 端付款結果通知後，請正確回應 1 OK 給綠界。 3. 1 OK 僅是特店回應綠界是否收到通知，並不會改變付款狀態。
	*TradeDesc	交易描述	String(200)	
	*ItemName	商品名稱	String(400)	商品名稱以分隔線#分開
CardInfo	信用卡資訊	JSON		以下情況為必填： 1. PaymentUIType 選擇 0 或 1 PaymentUIType 選擇 2，且 ChoosePaymentList 選擇 0、1 或 2。
Redeem	使用信用卡紅利	String (1)		預設為不使用 0: 不使用 1: 使用
PeriodAmount	定期定額每次授權金額	Int		當 PaymentUIType 為 0 時，此欄位必填
PeriodType	定期定額週期種類	String(1)		當 PaymentUIType 為 0 時，此欄位必填 D：以天為週期 M：以月為週期 Y：以年為週期
Frequency	定期定額執行頻率	Int		當 PaymentUIType 為 0 時，此欄位必填 注意事項： 至少要大於等於 1 次以上。 當 PeriodType 設為 D 時，最多可設 1~365(天)。 當 PeriodType 設為 M 時，最多可設 1~12(月)。 當 PeriodType 設為 Y 時，最多可設 1。
ExecTimes	定期定額執行次數	Int		當 PaymentUIType 為 0 時，此欄位必填。 注意事項： 至少要大於 1 次以上。 當 PeriodType 設為 D 時，最多可設 999 次。 當 PeriodType 設為 M 時，最多可設 99 次。 當 PeriodType 設為 Y 時，最多可設 9 次。
*OrderResultURL	3D 驗證回傳付款結果網址	String(200)		使用 3D 驗證時，當消費者付款完成後，綠界會將付款結果參數以前端頁面轉導的方式回傳到該網址。回傳參數請參考 付款結果通知章節
PeriodReturnURL	定期定額執行結果回應 URL	String(200)		當 PaymentUIType 為 0 時，此欄位必填若交易是信用卡定期定額的方式，則每次執行授權完，會將授權結果回傳到這個設定的 URL。
CreditInstallment	刷卡分期期數	String(20)		當 ChoosePaymentList 為 0 或 ChoosePaymentList 有選擇 2 時，此欄位必填支援多期數請以逗號分隔 ex: 3,6,12,18,24
TravelStartDate	國旅卡旅遊起日	String(8)		當 PaymentUIType 為 1 時，此欄位必填 格式為 mmddyyyy

TravelEndDate	國旅卡旅遊訖日	String(8)	當 PaymentUIType 為 1 時，此欄位必填 格式為 mmddyyyy
TravelCounty	國旅卡旅遊縣市	String(3)	當 PaymentUIType 為 1 時，此欄位必填 縣市群組名稱 縣市群組代碼 台北市 001 基隆市 002 新北市 003 宜蘭縣 004 新竹市 005 新竹縣 006 桃園縣 007 苗栗縣 008 台中市 009 彰化縣 011 南投縣 012 嘉義市 013 嘉義縣 014 雲林縣 015 台南市 016 高雄市 018 屏東縣 020 台東縣 021 花蓮縣 022 澎湖縣 023 金門縣 024 連江縣 025 南海諸島 026 釣魚台列嶼 027 屏東縣琉球鄉離島 028 台東縣綠島、蘭嶼 029
UnionPayInfo	銀聯卡資訊	JSON	當 PaymentUIType 為 0 或 6，此欄位必填
OrderResultURL	銀聯卡驗證回傳付款結果網址	String(200)	使用銀聯卡付款時，當消費者付款完成後，綠界會將付款結果參數以前端頁面轉導的方式回傳到該網址。回傳參數請參考 付款結果通知章節
ATMInfo	ATM 資訊	JSON	ChoosePaymentList 如選擇 0 或 3，則必填
*ExpireDate	允許繳費有效天數	Int	若需設定最長 60 天，最短 1 天。未設定此參數則預設為 3 天 注意事項： 以天為單位

	ATMBankCode	ATM 銀行代碼	String (10)	繳費銀行代碼 若未傳入，依系統預設銀行為主 請參考 銀行代碼表
CVSInfo		超商代碼資訊	JSON	ChoosePaymentList 如選擇 0 或 4，則必填
	*StoreExpireDate	超商繳費截止時間	Int	以分鐘為單位 若未設定此參數，預設為 10080 分鐘(7 天)。若需設定此參數，請於建立訂單時將此參數送給綠界。提醒您，CVS 帶入數值不可超過 86400 分鐘，超過時一律以 86400 分鐘計(60 天)例：08/01 的 20:15 分購買商品，繳費期限為 7 天，表示 08/08 的 20:15 分前您必須前往超商繳費。
	CVSCode	超商代碼	String(10)	CVS: 超商代碼繳款(不指定超商) OK: OK 超商代碼繳款 FAMILY: 全家超商代碼繳款 HILIFE: 萊爾富超商代碼繳款 IBON: 7-11 ibon 代碼繳款 若未傳入，預設為 CVS
	Desc_1	交易描述 1	String(20)	若繳費超商為 family(全家)或 ibon(7-11)時，會顯示在超商繳費平台螢幕上
	Desc_2	交易描述 2	String(20)	若繳費超商為 family(全家)或 ibon(7-11)時，會顯示在超商繳費平台螢幕上
	Desc_3	交易描述 3	String(20)	若繳費超商為 family(全家)或 ibon(7-11)時，會顯示在超商繳費平台螢幕上
	Desc_4	交易描述 4	String(20)	若繳費超商為 family(全家)或 ibon(7-11)時，會顯示在超商繳費平台螢幕上
BarcodeInfo		超商條碼資訊	JSON	ChoosePaymentList 如選擇 0 或 5，則必填
	*StoreExpireDate	超商繳費截止時間	Int	以天為單位，預設為 7 天。
ConsumerInfo		特店會員資訊	JSON	當特店擁有自己的會員系統時，可將會員資料帶入
	MerchantMemberID	特店會員編號	String(60)	當[RememberCard] = 1，此欄位必填 請帶入特店的會員編號
	Email	信用卡持卡人電子信箱	String(30)	
	Phone	信用卡持卡人電話	String(60)	注意事項： 可接受帶國碼的形式，但不可帶+號 ex: 886912345678
	Name	信用卡持卡人姓名	String(50)	注意事項： 該欄位可接受中文、英文與部分特殊符號。 可支援的特殊符號為: ,./()-

CountryCode	國別碼	String(3)	持卡人帳單地址國別碼，請參考 ISO3166 。臺灣請填寫 158
Address	地址	String(50)	持卡人帳單地址
CustomField	特店自訂欄位	String(200)	提供特店使用客制化欄位

Data 範例：

```
{
  "GatewayMerchantID": "3002607",
  "RememberCard": 1,
  "PaymentUIType": 2,
  "ChoosePaymentList": "1,2,3",
  "OrderInfo": {
    "MerchantTradeNo": "20180914001",
    "MerchantTradeDate": "2020/09/26 14:49:12",
    "TotalAmount": 100,
    "ReturnURL": "https://yourReturnURL.com",
  },
  "CardInfo": {
    "OrderResultURL": "https://yourOrderResultURL.com",
    "CreditInstallment": "3,6,9,12"
  },
  "ATMInfo": {
    "ExpireDate": 3
  },
  "ConsumerInfo": {
    "MerchantMemberID": "test123456",
    "Email": "customer@email.com",
    "Phone": "0912345678",
    "Name": "Test",
    "CountryCode": "158"
  }
}
```

● 綠界回傳參數說明：(JSON 格式)

參數	參數名稱	型態	說明
MerchantID	特店編號	String(10)	特店編號
RpHeader	回傳資料	JSON	

	Timestamp	回傳時間	Number	時間戳 Unix timestamp
TransCode	回傳代碼	Int	1 代表 API 傳輸資料(MerchantID, RqHeader, Data)接收成功，實際的 API 執行結果狀態請參考 RtnCode	
TransMsg	回傳訊息	String(200)	回傳訊息	
Data	加密資料	String	加密過 JSON 格式的資料。 加密方法說明	

範例:

```
{
  "MerchantID": "3002607",
  "RpHeader": {
    "Timestamp": 1234564848
  },
  "TransCode": 1,
  "TransMsg": "Success",
  "Data": "..."
}
```

- Data 參數說明：請在解密後對參數值進行 URLDecode

參數	參數名稱	型態	說明
RtnCode	交易狀態	Int	1 代表 API 執行成功，代表取得廠商驗證碼成功。其餘代碼均為失敗。失敗代碼請參考 交易訊息代碼表
RtnMsg	回應訊息	String(200)	
RtnCode 為 1 時，以下參數才会有值:			
PlatformID	平台商編號	String(10)	特約合作平台商特店編號。
MerchantID	特店編號	String(10)	
Token	交易代碼	String(64)	可使用的效期為 30 分鐘
TokenExpireDate	交易代碼到期時間	String(20)	格式為 yyyy/MM/dd HH:mm:ss

Data 解密後:

```
{  
  "RtnCode": 1,  
  "RtnMsg": "Success",  
  "PlatformID": "1234567890",  
  "MerchantID": "1234567890",  
  "Token": "m12dae4846446sq",  
  "TokenExpireDate": "2020/09/18 15:39:10"  
}
```

(2/2)

(2) 刪除記憶信用卡資料

- 應用場景：使用者可於交易中進行記憶卡號功能，當使用者想要刪除信用卡資料，特店可透過此功能刪除已綁定的信用卡。
- 介接路徑
 - 正式環境：https://ecpg.ecpay.com.tw/Merchant/GetTokenbyUser
 - 測試環境：https://ecpg-stage.ecpay.com.tw/Merchant/GetTokenbyUser
- 特店傳入參數說明：(JSON 格式，名稱若有紅色星號*為必填欄位)
 - Content Type : application/json
 - HTTP Method : POST

參數	參數名稱	型態	說明
*MerchantID	特店編號	String(10)	特店編號 若為平台商時，參數請帶平台商所綁的特店編號
*RqHeader	傳輸資料	JSON	
	*Timestamp	Number	請將傳輸時間轉換為時間戳(GMT+8)，綠界會利用此參數將當下的時間轉為 Unix TimeStamp 來驗證此次介接的時間區間。 注意事項： 若時間戳跟綠界伺服器接收到時間超過 10 分鐘時，交易會失敗無法進行
	*Revision	String(10)	串接文件版號。Ex: 1.0.0
*Data	加密資料	String	加密過 JSON 格式的資料。 加密方法說明

範例:

```
{
  "MerchantID": "3002607",
  "RqHeader": {
    "Timestamp": 1525168923,
    "Revision": "1.0.0"
  },
  "Data": "..."
}
```

● Data 參數說明：請在加密前對參數值進行 URLEncode

參數	參數名稱	型態	說明
PlatformID	特約合作平台商代號	String(10)	
*MerchantID	特店編號	String(10)	
*ConsumerInfo	特店會員資訊	JSON	
*MerchantMemberID	特店會員編號	String(60)	
Email	信用卡持卡人電子信箱	String(30)	
Phone	信用卡持卡人電話	String(60)	
Name	信用卡持卡人姓名	String(50)	
CountryCode	國別碼	String(3)	持卡人帳單地址國別碼・IOS3166
Address	地址	String(50)	持卡人帳單地址
CustomField	特店自訂欄位	String(200)	提供特店使用客制化欄位

Data 範例：

```
{
  "MerchantID": "3002607",
  "ConsumerInfo": {
    "MerchantMemberID": "test123466",
    "Email": "customer@email.com",
    "Phone": "0912345678",
    "Name": "Test",
    "CountryCode": "158"
  },
  "ReturnURL": "https://aa.aa.aa"
}
```

● 綠界回傳參數說明：(JSON 格式)

參數	參數名稱	型態	說明
MerchantID	特店編號	String(10)	特店編號
RpHeader	回傳資料	JSON	
Timestamp	回傳時間	Int	時間戳 Unix timestamp
TransCode	回傳代碼	Int	1 代表 API 傳輸資料(MerchantID, RqHeader, Data)接收成功・實際的 API 執行結果狀態請參考 RtnCode
TransMsg	回傳訊息	String(200)	回傳訊息

Data	加密資料	String	加密過 JSON 格式的資料。 加密方法說明
------	------	--------	--

範例:

```
{
  "MerchantID": "3002607",
  "RpHeader": {
    "Timestamp": 1525169058
  },
  "TransCode": 1,
  "TransMsg": "Success",
  "Data": "..."
}
```

- Data 參數說明：請在解密後對參數值進行 URLDecode

參數	參數名稱	型態	說明
RtnCode	交易狀態	Int	1 代表 API 執行成功，代表取得驗證碼成功。其餘代碼均為失敗。失敗代碼請參考 交易訊息代碼表
RtnMsg	回應訊息	String(200)	
RtnCode 為 1 時，以下參數才會有值：			
PlatformID	平台商編號	String(10)	特約合作平台商特店編號。
MerchantID	特店編號	String(10)	
Token	廠商驗證碼	String(64)	效期為 30 分鐘
TokenExpireDate	廠商驗證碼到期時間	String(20)	yyyy/MM/dd HH:mm:ss

Data 解密後:

```
{
  "RtnCode": 1,
  "RtnMsg": "Success",
  "PlatformID": "1234567890",
  "MerchantID": "1234567890",
  "Token": "m12dae4846446sq",
  "TokenExpireDate": "2020/09/18 15:39:10"
}
```

5. 一般交易嵌入式畫面 SDK(JS)

- 應用場景:

適用於前端 Web，提供特店站內付 2.0 金流功能，詳細用法請參考[範例程式](#)。

- 使用方式:

1. 引用外部 JS 元件:

```
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
```

2. 引用綠界 JS 元件:

- 正式環境：

```
<script src="https://ecpg.ecpay.com.tw/Scripts/sdk-1.0.0.js?t=20210121100116"></script>
```

- 測試環境：

```
<script src="https://ecpg-stage.ecpay.com.tw/Scripts/sdk-1.0.0.js?t=20210121100116"></script>
```

注意事項:

(1) 引用綠界 JS 元件時，必須放在<div id="ECPayPayment"></div>的後面

3. Html 區塊定義:

```
<div id="ECPayPayment"></div> //呈現付款方式畫面
```

注意事項:

(1) 定義 CSS 參數時，請勿與站內付 2.0 SDK 定義的相同，請[參考 CSS 參數定義列表](#)。

4. 初始化 SDK，環境參數請參考下段說明。

```
ECPay.initialize( ServerType, IsLoading, callBack(errMsg) )
```

■ 特店帶入參數:

參數	參數名稱	型態	說明
*ServerType	環境參數	String	Stage:測試環境 ECPay.ServerType.Stage Prod:正式環境 ECPay.ServerType.Prod
IsLoading	是否顯示等待畫面	Int	0:不顯示等待畫面 1:顯示等待畫面 如無設定該值，預設為 1
*callBack	CallBack	Function	執行回呼函式 1.errMsg - 執行失敗時，可在 callBack 接收錯誤訊息 2.需於此處呼叫 ECPay.createPayment

5. 取得一般交易付款畫面

ECPay.createPayment(Token, Language, callBack(errMsg))

■ 特店帶入參數:

參數	參數名稱	型態	說明
*Token	廠商驗證碼	String(64)	初始化站內付 2.0 畫面代碼 此參數具 30 分鐘時效性
*Language	多語系	String(10)	可使用以下方式設定使用語系 繁中: ECPay.Language.zhTW 英語: ECPay.Language.enUS
*callback	Callback	Function	errMsg - 執行失敗時，可在 callBack 接收錯誤訊息，成功時該參數為 null

6. 取得付款代碼(PayToken)

ECPay.getPayToken(callBack(paymentInfo, errMsg))

■ 綠界回傳參數:

參數	參數名稱	型態	說明
PaymentInfo	付款資訊	JSON	
PayToken	付款代碼	String(64)	付款代碼; 有效期限為 30 分鐘 取得付款代碼後，用法請參考 章節 6.建立交易
PaymentType	使用者選擇的付款方式	String(20)	CreditCard:信用卡一次付清 CreditInstallment:信用卡分期付款 ATM: ATM

			CVS: 超商代碼 Barcode: 超商條碼 Period: 定期定額 UnionPay: 銀聯卡 ApplePay: Apple Pay ※注意事項: 當 PaymentType 為 Apple Pay 時，不會回傳 PayToken。特店必須實作 GetApplePayResultData 直接取得 Apple Pay 付款結果。
callBack	callBack	Function	1.paymentInfo - 執行成功時，可取得付款設定 2.errMsg - 執行失敗時，可在 callBack 接收錯誤訊息

7. GetLanguage: 取得設定語系

`ECPay.getLanguage()`

■ 綠界回傳參數:

內容	說明
zh-TW · en-US	切換語系時，可取出於 SDK 目前設定的語系(預設:zh-TW)

8. GetApplePayResultData: 取得 ApplePay 付款結果

`getApplePayResultData( resultData, errMsg )`

注意事項:

- (1) 如特店支援的付款方式有 Apple Pay 功能，必須自行實作此 function 接收 Apple Pay 付款結果。如無實做，會無法收到付款結果。
- (2) 實際的付款結果請依照付款結果通知參數 [ReturnURL](#) 收到的回傳結果為主。
- (3) 特店可使用收到的特店編號[MerchantID]與特店交易編號[MerchantTradeNo]，利用[查詢訂單 API](#)查詢是否為有效訂單。

■ 綠界回傳參數:

參數	參數名稱	型態	說明
RtnCode	交易狀態	Int	1 代表 API 執行成功，其餘代碼均為失敗。 失敗代碼請參考 交易訊息代碼表
RtnMsg	回應訊息	String(200)	
以下為 RtnCode 為 1 時才會回傳			
PlatformID	平台商編號	String(10)	特約合作平台商特店編號。
MerchantID	特店編號	String(10)	

OrderInfo		訂單資訊	JSON Object	
	MerchantTradeNo	特店交易編號	String(20)	
	TradeNo	綠界的交易編號	String(20)	請保存綠界的交易編號與特店交易編號[MerchantTradeNo]的關聯。
	PaymentDate	付款時間	String(20)	格式為 yyyy/MM/dd HH:mm:ss
	TradeAmt	交易金額	Int	
	PaymentType	特店選擇的付款方式	String(20)	
	TradeDate	訂單成立時間	String(20)	yyyy/MM/dd HH:mm:ss
	ChargeFee	手續費	Number	
	TradeStatus	交易狀態	String(8)	回傳值： 若為 0 時，代表交易訂單成立未付款 若為 1 時，代表交易訂單成立已付款
CardInfo		授權資訊	JSON Object	
	AuthCode	銀行授權碼	String(6)	
	Gwsr	授權交易單號	Int	
	ProcessDate	交易時間	String(20)	yyyy/MM/dd HH:mm:ss
	Amount	金額	Int	
	Eci	3D(VBV) 回傳值	Int	
	Card6No	信用卡卡號前六碼	String(6)	
	Card4No	信用卡卡號末四碼	String(4)	
	CustomField	特店自訂欄位	String(200)	提供特店使用客制化欄位

● 範例程式:

```
//初始化 SDK
ECPay.initialize(serverType, function (errMsg) {
    var lang = getCookie("language");
    if (!lang) {
        lang = ECPay.Language.zhTW;
    }
    ECPay.createPayment("55a026b8df314f9bb2dc808a5fe21c9d", lang, function (errMsg) {
        });
    if ($(this).val() === ECPay.getLanguage()) {
        $(this).get(0).selected = true;
        return false;
    };
    function getApplePayResultData(resultData, errMsg) {
        Console.log(resultData);
        If(resultData != null){
            /*請依實際需求自行處理回傳結果*/
        }
    };
});
```


6. 建立一般交易(Server)

- 應用場景：

特店 Server 需將從特店 Web 取得到的 PayToken，送至綠界 Server 建立一般交易。

- 介接路徑
 - 正式環境：<https://ecpg.ecpay.com.tw/Merchant/CreatePayment>
 - 測試環境：<https://ecpg-stage.ecpay.com.tw/Merchant/CreatePayment>
- 廠商傳入參數說明：(JSON 格式)
 - Content Type : application/json
 - HTTP Method : POST

參數	參數名稱	型態	說明
*MerchantID	特店編號	String(10)	特店編號 若為平台商時，參數請帶平台商所綁的特店編號
*RqHeader	傳輸資料	JSON	
*Timestamp	傳輸時間	Number	請將傳輸時間轉換為時間戳(GMT+8)，綠界會利用此參數將當下的時間轉為 Unix TimeStamp 來驗證此次介接的時間區間。 注意事項： 若時間戳跟綠界伺服器接收到時間超過 10 分鐘時，交易會失敗無法進行
*Revision	串接文件版號	String(10)	請參考文件封面的 文件版號 。Ex: 1.0.0
*Data	加密資料	String	加密過 JSON 格式的資料。 加密方法說明

範例:

```
{
  "MerchantID": "3002607",
  "RqHeader": {
    "Timestamp": 1234567890,
    "Revision": "1.0.0"
  },
  "Data": "..."
}
```

- Data 參數說明：請在加密前對參數值進行 URLEncode

參數	參數名稱	型態	說明
PlatformID	特約合作平台商代號	String(10)	
*GatewayMerchantID	閘道商特店編號	String(10)	
*PayToken	付款代碼	String(64)	
*MerchantTradeNo	特店交易編號	String(20)	必須與 取得廠商驗證碼 API 的 MerchantTradeNo 相同

Data 範例：

```
{
  "GatewayMerchantID": "3002607",
  "PayToken": "enter your paytoken",
  "MerchantTradeNo": "test123456789"
}
```

- 綠界回傳參數說明：回傳 3D URL 或授權結果

參數	參數名稱	型態	說明
MerchantID	特店編號	String(10)	特店編號
RpHeader	回傳資料	JSON	
Timestamp	回傳時間	Number	時間戳 Unix timestamp
TransCode	回傳代碼	Int	1 代表 API 傳輸資料(MerchantID, RqHeader, Data)接收成功，實際的 API 執行結果狀態請參考 RtnCode
TransMsg	回傳訊息	String(200)	回傳訊息
Data	加密資料	String	加密過 JSON 格式的資料。 加密方法說明

範例:

```
{
  "MerchantID": "3002607",
  "RpHeader": {
    "Timestamp": 1234567890
  },
  "TransCode": 1,
  "TransMsg": "Success",
  "Data": "..."
}
```

● Data 參數說明：需進行 3D 驗證時，回傳以下參數

參數	參數名稱	型態	說明
RtnCode	交易狀態	Int	1 代表 API 執行成功，代表建立交易成功。其餘代碼均為失敗。失敗代碼請參考 交易訊息代碼表
RtnMsg	回應訊息	String(200)	
以下為 RtnCode 為 1 時才會回傳			
PlatformID	平台商編號	String(10)	特約合作平台商特店編號。
MerchantID	特店編號	String(10)	
OrderInfo	訂單資訊	JSON Object	
MerchantTradeNo	特店交易編號	String(20)	
ThreeDInfo	3D 驗證資訊	JSON Object	
ThreeDURL	3D 驗證 URL	String(200)	3D 驗證連結，請勿使用 iframe 方式開啟。

Data 解密後：

```
{
  "RtnCode": 1,
  "RtnMsg": "Success",
  "PlatformID": "1234567890",
  "MerchantID": "1234567890",
  "OrderInfo": {
    "MerchantTradeNo": "test123466"
  },
  "ThreeDInfo": {
    "ThreeDURL": "https://3durl.com.tw"
  }
}
```

- Data 參數說明：需進行銀聯卡驗證時，回傳以下參數

參數	參數名稱	型態	說明
RtnCode	交易狀態	Int	1 代表 API 執行成功，代表建立交易成功。其餘代碼均為失敗。失敗代碼請參考 交易訊息代碼表
RtnMsg	回應訊息	String(200)	
以下為 RtnCode 為 1 時才會回傳			
PlatformID	平台商編號	String(10)	特約合作平台商特店編號。
MerchantID	特店編號	String(10)	
OrderInfo	訂單資訊	JSON Object	
MerchantTradeNo	特店交易編號	String(20)	
UnionPayInfo	銀聯卡驗證資訊	JSON Object	
UnionPayURL	銀聯卡驗證 URL	String(200)	※注意事項： 1. 銀聯卡驗證連結，請勿使用 iframe 方式開啟。 2. 測試環境提供的驗證 URL 為綠界模擬測試頁面。

Data 解密後：

```
{
  "RtnCode": 1,
  "RtnMsg": "Success",
  "PlatformID": "1234567890",
  "MerchantID": "1234567890",
  "OrderInfo": {
    "MerchantTradeNo": "test123466"
  },
  "UnionPayInfo": {
    "UnionPayURL": "https://unionpayurl.com.tw"
  }
}
```

● Data 參數說明：不需 3D 驗證，直接回傳授權結果

參數	參數名稱	型態	說明
RtnCode	交易狀態	Int	1 代表 API 執行成功，代表建立交易成功。其餘代碼均為失敗。失敗代碼請參考 交易訊息代碼表
RtnMsg	回應訊息	String(200)	
以下為 RtnCode 為 1 時才會回傳			
PlatformID	平台商編號	String(10)	特約合作平台商特店編號。
MerchantID	特店編號	String(10)	
OrderInfo	訂單資訊	JSON Object	
MerchantTradeNo	特店交易編號	String(20)	
TradeNo	綠界交易編號	String(20)	
TradeAmt	交易金額	Int	
TradeDate	訂單成立時間	String(20)	yyyy/MM/dd HH:mm:ss
PaymentType	付款方式	String(20)	
PaymentDate	付款時間	String(20)	yyyy/MM/dd HH:mm:ss
ChargeFee	手續費	Number	
TradeStatus	交易狀態	String(8)	
CardInfo	信用卡授權資訊	JSON Object	
AuthCode	銀行授權碼	String(6)	
Gwsr	授權交易單號	Int	
ProcessDate	交易時間	String(20)	yyyy/MM/dd HH:mm:ss
Amount	金額	Int	
Stage	分期期數	Int	
Stast	首期金額	Int	
Staed	各期金額	Int	
Eci	3D(VBV) 回傳值	Int	Eci=5,6,2,1 代表該筆交易為 3D 交易
Card6No	信用卡卡號前六碼	String(6)	
Card4No	信用卡卡號末四碼	String(4)	
RedDan	紅利扣點	Int	使用信用卡紅利時回傳
RedDeAmt	紅利折抵金額	Int	使用信用卡紅利時回傳
RedOkAmt	實際扣款金額	Int	使用信用卡紅利時回傳
RedYet	紅利剩餘點數	Int	使用信用卡紅利時回傳

PeriodType	週期種類	String (1)	定期定額時回傳，訂單建立時所設定的週期種類
Frequency	執行頻率	Int	定期定額時回傳，訂單建立時所設定的執行頻率
ExecTimes	執行次數	Int	定期定額時回傳，訂單建立時所設定的執行頻率
PeriodAmount	訂單建立時的每次要授權金額	Int	定期定額時回傳
TotalSuccessTimes	目前已成功授權的次數	Int	定期定額時回傳，目前已成功授權的次數。
TotalSuccessAmount	目前已成功授權的金額合計	Int	定期定額時回傳
ATMInfo		ATM 資訊	JSON Object
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
CVSInfo		超商代碼資訊	JSON Object
PaymentNo	繳費代碼	String(14)	
ExpireDate	繳費期限	String(20)	格式為 yyyy/MM/dd HH:mm:ss
PaymentURL	繳費連結	String(100)	由綠界科技提供手機上顯示的三段式繳費條碼網頁
BarcodeInfo		超商條碼資訊	JSON Object
ExpireDate	繳費期限	String(20)	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
Barcode2	條碼第二段號碼	String(20)	格式為 16 碼數字
Barcode3	條碼第三段號碼	String(20)	格式為 15 碼數字
CoBrandingInfo		聯名卡資料	Array
CoBrandingCode	聯名卡代號	String(20)	綠界廠商後台設定的聯名卡代號
Comment	備註	String(100)	
CustomField	特店自訂欄位	String(200)	提供特店使用客制化欄位

Data 解密後:

```
{
  "RtnCode": 1,
  "RtnMsg": "Success",
  "PlatformID": "1234567890",
  "MerchantID": "1234567890",
  "OrderInfo": {
    "MerchantTradeNo": "test123466",

```

```
"TradeNo": "123456789",  
"TradeDate": "2020/09/18 15:39:10"  
},  
"CardInfo": {  
  "AuthCode": "123456",  
  "Gwsr": 12546,  
  "ProcessDate": "2020/09/18 15:39:10",  
  "Amount": 500,  
  "Stage": 1,  
  "Stast": 500,  
  "Staed": 500,  
  "Eci": 9,  
  "Card6No": "123456",  
  "Card4No": "1234",  
  "RedDan": 10,  
  "RedDeAmt": 10,  
  "RedOkAmt": 490,  
  "RedYet": 0  
}  
}
```

7. 定期定額查詢(Server)

- 應用場景：

提供查詢信用卡定期定額明細記錄。

- 介接路徑：

- 正式環境：<https://ecpayment.ecpay.com.tw/1.0.0/Cashier/QueryTrade>

- 測試環境：<https://ecpayment-stage.ecpay.com.tw/1.0.0/Cashier/QueryTrade>

- 特店傳入參數說明：(JSON 格式)

- Content Type : application/json

- HTTP Method : POST

參數	參數名稱	型態	說明
*MerchantID	特店編號	String(10)	特店編號 若為平台商時，參數請帶平台商所綁的特店編號
*RqHeader	傳輸資料	JSON	
*Timestamp	傳輸時間	Number	請將傳輸時間轉換為時間戳(GMT+8)，綠界會利用此參數將當下的時間轉為 Unix TimeStamp 來驗證此次介接的時間區間。 注意事項： 若時間戳跟綠界伺服器接收到時間超過 10 分鐘時，交易會失敗無法進行
*Revision	串接版號	String(10)	請參考文件封面的 文件版號 。Ex: 1.0.0
*Data	加密資料	String	加密過 JSON 格式的資料。 加密方法說明

範例:

```
{
  "MerchantID": "3002607",
  "RqHeader": {
    "Timestamp": 1234567890,
    "Revision": "1.0.0"
  },
  "Data": "..."
}
```


- Data 參數說明：請在加密前對參數值進行 URLEncode

參數	參數名稱	型態	說明
PlatformID	平台商編號	String(10)	特約合作平台商特店編號
*MerchantID	特店編號	String(10)	
*MerchantTradeNo	特店交易編號	String(20)	

Data 範例：

```
{
  "MerchantID": "3002607",
  "PlatformID": "2000132",
  "MerchantTradeNo": "123486456"
}
```

- 綠界回傳參數說明：(JSON 格式)

參數	參數名稱	型態	說明
MerchantID	特店編號	String(10)	特店編號
RpHeader	傳輸資料	JSON	
	Timestamp	Number	時間戳 Unix timestamp 注意事項: 若時間戳跟綠界伺服器接收到時間超過 10 分鐘時，交易會失敗無法進行
TransCode	回傳代碼	Int	1 代表 API 傳輸資料(MerchantID, RpHeader, Data)接收成功，實際的 API 執行結果狀態請參考 RtnCode
TransMsg	回傳訊息	String(200)	回傳訊息
Data	加密資料	String	加密過 JSON 格式的資料。 加密方法說明

範例:

```
{
  "MerchantID": "3002607",
  "RpHeader": {
    "Timestamp": 1525169058
  },
  "TransCode": 1,
  "TransMsg": "Success",
  "Data": "..."
}
```

● Data 參數說明：請在解密後對參數值進行 URLDecode

參數	參數名稱	型態	說明
RtnCode	交易狀態	Int	1 為成功，其餘為失敗
RtnMsg	交易訊息	String(200)	
RtnCode 為 1 時，以下參數才有值：			
PlatformID	特約合作平台商代號	String(10)	
MerchantID	特店編號	String(10)	
OrderInfo	訂單資訊	JSON	此欄位顯示為定期定額的第一筆資料，第二筆之後的資料為 ExecLog
MerchantTradeNo	特店交易編號	String (20)	
TradeNo	綠界的交易編號	String (20)	
PaymentDate	付款時間	String (20)	yyyy/MM/dd HH:mm:ss
TradeAmt	交易金額	Int	
PaymentType	付款方式	String (20)	Credit/CVS/ATM/BARCODE
TradeDate	訂單時間	String (20)	yyyy/MM/dd HH:mm:ss
ChargeFee	手續費	Number	
TradeStatus	交易狀態	String (8)	回傳值： 若為 0 時，代表交易訂單成立未付款 若為 1 時，代表交易訂單成立已付款
CardInfo	信用卡授權資訊	JSON	
AuthCode	銀行授權碼	String(6)	
Gwsr	授權交易單號	Int	
ProcessDate	交易時間	String(20)	yyyy/MM/dd HH:mm:ss
Amount	金額	Int	
Eci	3D(VBV) 回傳值	Int	Eci=5,6,2,1 代表該筆交易為 3D 交易
Card6No	信用卡卡號前六碼	String(6)	
Card4No	信用卡卡號末四碼	String(4)	
PeriodType	週期種類	String(1)	
Frequency	執行頻率	Int	
ExecTimes	執行次數	Int	
PeriodAmount	訂單建立時的每次要授權金額	Int	
TotalSuccessTimes	已執行成功次數	Int	目前已成功授權的次數。
TotalSuccessAmount	目前已成功授權的金	Int	

		額合計		
ExecStatus	執行狀態	String(1)	回傳值如下： 1.若為 0 時代表已取消 2.若為 1 時代表執行中 3.若為 2 時代表執行完成	
ExecLog	每次授權明細	Array	每一次的授權紀錄。 例如，若已成功授權 3 次(TotalSuccessTimes 為 3)，則會顯示 3 筆授權紀錄。	
RtnCode	交易狀態	Int	若回傳值時為 1 時代表授權成功，其餘為失敗，失敗代碼請參考交易訊息代碼一覽表	
Amount	授權金額	Int		
Gwsr	授權交易單號	Int		
ProcessDate	授權成功處理時間	String(20)	格式為 yyyy/MM/dd HH:mm:ss	
AuthCode	授權碼	String(6)		
TradeNo	綠界的交易編號	String(20)	請保存綠界的交易編號與特店交易編號 [MerchantTradeNo]的關聯。	
ChargeFee	手續費	Number		
CustomField	特店自訂欄位	String(200)	提供特店使用客制化欄位	

Data 解密後：

```

{
  "RtnCode": 1,
  "RtnMsg": "Success",
  "MerchantID": "3002607",
  "OrderInfo": {
    "MerchantTradeNo": "20180914001",
    "TradeNo": "1809261503338172",
    "PaymentDate": "2018/09/26 14:59:54",
    "TradeAmt": 500,
    "PaymentType": "Credit",
    "TradeDate": "2018/09/26 14:59:54",
    "ChargeFee": 10,
    "TradeStatus": "1"
  },
  "CardInfo": {
    "AuthCode": "777777",
    "Gwsr": 10735183,

```

```
"ProcessDate": "2018/09/26 14:59:54",
"Amount": 100,
"Eci": 2,
"Card4No": "2222",
"Card6No": "491122",
"Frequency": 2,
"ExecTimes": 1,
"PeriodAmount": 2,
"TotalSuccessTimes": 3,
"TotalSuccessAmount": 5
},
"ExecStatus": "1",
"ExecLog": {
    "RtnCode": 1,
    "Amount": 500,
    "Gwsr": 12345,
    "ProcessDate": "2020/12/31 15:12:20",
    "AuthCode": "123456",
    "TradeNo": "12345649781",
    "ChargeFee": 10
}
```

8. 信用卡定期定額訂單作業(Server)

- 應用場景：

提供特店執行信用卡定期定額訂單交易作業。

- ReAuth：最新一筆定期定額交易授權失敗時，可發動進行補授權作業。

- Cancel：停用本筆定期定額後續授權交易作業。

- 介接路徑：

- 正式環境：<http://ecpayment.ecpay.com.tw/1.0.0/Cashier/CreditCardPeriodAction>

- 測試環境：<http://ecpayment-stage.ecpay.com.tw/1.0.0/Cashier/CreditCardPeriodAction>

- 特店傳入參數說明：(JSON 格式)

- Content Type : application/json

- HTTP Method : POST

參數	參數名稱	型態	說明
*MerchantID	特店編號	String(10)	特店編號 若為平台商時，參數請帶平台商所綁的特店編號
*RqHeader	傳輸資料	JSON	
*Timestamp	傳輸時間	Number	請將傳輸時間轉換為時間戳(GMT+8)，綠界會利用此參數將當下的時間轉為 Unix TimeStamp 來驗證此次介接的時間區間。 注意事項： 若時間戳跟綠界伺服器接收到時間超過 10 分鐘時，交易會失敗無法進行
*Revision	串接版號	String(10)	請參考文件封面的 文件版號 。Ex: 1.0.0
*Data	加密資料	String	加密過 JSON 格式的資料。 加密方法說明

範例:

```
{
  "MerchantID": "3002607",
  "RqHeader": {
    "Timestamp": 1525168923,
    "Revision": "1.0.0"
  },
}
```

```
"Data": "..."  
}
```

- Data 參數說明：請在加密前對參數值進行 URLEncode

參數	參數名稱	型態	說明
PlatformID	平台商編號	String(10)	特約合作平台商特店編號
*MerchantID	特店編號	String(10)	
*MerchantTradeNo	特店交易編號	String(20)	
*Action	作業方式	String(20)	ReAuth：補授權失敗交易 Cancel：停用定期定額後續交易

注意事項:

- (1) 定期訂額信用卡補授權作業，限當筆交易最新一筆授權失敗時才能使用，例如第二期授權失敗，但第三期成功時，此時無法補授權第二期失敗的交易。
- (2) 停用交易成功後，無法重新啟用，只能重新發動新定期定額訂單進行交易

Data 範例：

```
{  
  "MerchantID": "3002607",  
  "PlatformID": "2000132",  
  "MerchantTradeNo": "123486456",  
  "Action": "ReAuth"  
}
```

- 綠界回傳參數說明：(JSON 格式)

參數	參數名稱	型態	說明
MerchantID	特店編號	String(10)	特店編號
RpHeader	傳輸資料	JSON	
Timestamp	傳輸時間	Number	時間戳 Unix timestamp 注意事項: 若時間戳跟綠界伺服器接收到時間超過 10 分鐘時，交易會失敗無法進行
TransCode	回傳代碼	Int	1 代表 API 傳輸資料(MerchantID, RqHeader, Data)接收成功，實際的 API 執行結果狀態請參考 RtnCode
TransMsg	回傳訊息	String(200)	回傳訊息
Data	加密資料	String	加密過 JSON 格式的資料。 加密方法說明

範例:

```
{
  "MerchantID": "3002607",
  "RpHeader": {
    "Timestamp": 1525169058
  },
  "TransCode": 1,
  "TransMsg": "Success",
  "Data": "..."
}
```

- Data 參數說明：請在解密後對參數值進行 URLDecode

參數	參數名稱	型態	說明
RtnCode	交易狀態	Int	1 為成功，其餘為失敗
RtnMsg	交易訊息	String(200)	
RtnCode 為 1 時，以下參數才会有值:			
PlatformID	特約合作平台商代號	String(10)	
MerchantID	特店編號	String(10)	
MerchantTradeNo	特店交易編號	String(20)	

Data 解密後：

```
{
  "RtnCode": 1,
  "RtnMsg": "Success",
  "MerchantID": "3002607",
  "MerchantTradeNo": "123456789014"
}
```

9. 查詢 ATM/CVS/BARCODE 取號結果(Server)

- 應用場景：

提供查詢 ATM/CVS/BARCODE 取號結果。

- 介接路徑：

- 正式環境：<http://ecpayment.ecpay.com.tw/1.0.0/Cashier/QueryPaymentInfo>
- 測試環境：<http://ecpayment-stage.ecpay.com.tw/1.0.0/Cashier/QueryPaymentInfo>

- 特店傳入參數說明：(JSON 格式)

- Content Type : application/json
- HTTP Method : POST

參數	參數名稱	型態	說明
*MerchantID	特店編號	String(10)	特店編號 若為平台商時，參數請帶平台商所綁的特店編號
*RqHeader	傳輸資料	JSON	
*Timestamp	傳輸時間	Number	請將傳輸時間轉換為時間戳(GMT+8)，綠界會利用此參數將當下的時間轉為 Unix TimeStamp 來驗證此次介接的時間區間。 注意事項： 若時間戳跟綠界伺服器接收到時間超過 10 分鐘時，交易會失敗無法進行
*Revision	串接版號	String(10)	請參考文件封面的 文件版號 。Ex: 1.0.0
*Data	加密資料	String	加密過 JSON 格式的資料。 加密方法說明

範例:

```
{
  "MerchantID": "3002607",
  "RqHeader": {
    "Timestamp": 1525168923,
    "Revision": "1.0.0"
  },
  "Data": "..."
}
```


- Data 參數說明：請在加密前對參數值進行 URLEncode

參數	參數名稱	型態	說明
PlatformID	平台商編號	String(10)	特約合作平台商特店編號
*MerchantID	特店編號	String(10)	
*MerchantTradeNo	特店交易編號	String(20)	

Data 範例：

```
{
  "MerchantID": "3002607",
  "PlatformID": "2000132",
  "MerchantTradeNo": "123486456"
}
```

- 綠界回傳參數說明：(JSON 格式)

參數	參數名稱	型態	說明
MerchantID	特店編號	String(10)	特店編號
RpHeader	傳輸資料	JSON	
Timestamp	傳輸時間	Number	時間戳 Unix timestamp 注意事項: 若時間戳跟綠界伺服器接收到時間超過 10 分鐘時，交易會失敗無法進行
TransCode	回傳代碼	Int	1 代表 API 傳輸資料(MerchantID, RpHeader, Data)接收成功，實際的 API 執行結果狀態請參考 RtnCode
TransMsg	回傳訊息	String(200)	回傳訊息
Data	加密資料	String	加密過 JSON 格式的資料。 加密方法說明

範例:

```
{
  "MerchantID": "3002607",
  "RpHeader": {
    "Timestamp": 1525169058
  },
  "TransCode": 1,
  "TransMsg": "Success",
  "Data": "..."
}
```

● Data 參數說明：請在解密後對參數值進行 URLDecode

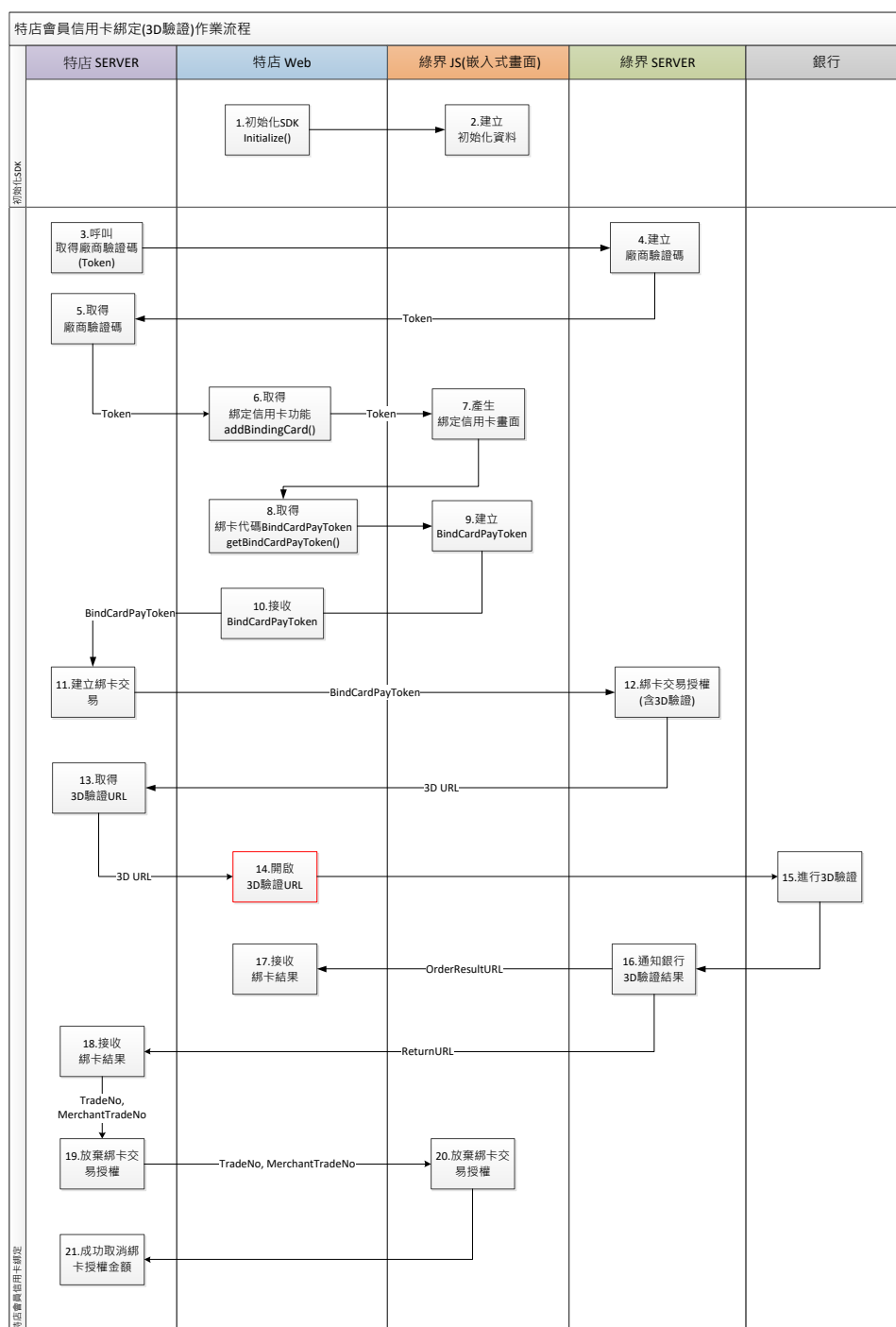
參數		參數名稱	型態	說明
RtnCode		交易狀態	Int	1 為成功，其餘為失敗
RtnMsg		交易訊息	String(200)	
RtnCode 為 1 時，以下參數才有值：				
PlatformID		特約合作平台商代號	String(10)	
MerchantID		特店編號	String(10)	
OrderInfo		訂單資訊	JSON	
	MerchantTradeNo	特店交易編號	String (20)	
	TradeNo	綠界的交易編號	String (20)	
	PaymentDate	付款時間	String (20)	yyyy/MM/dd HH:mm:ss
	TradeAmt	交易金額	Int	
	PaymentType	付款方式	String (20)	Credit/CVS/ATM/BARCODE
	TradeDate	訂單時間	String (20)	yyyy/MM/dd HH:mm:ss
	ChargeFee	手續費	Number	
	TradeStatus	交易狀態	String (8)	回傳值： 若為 0 時，代表交易訂單成立未付款 若為 1 時，代表交易訂單成立已付款
CVSInfo		超商代碼資訊	JSON	
	PaymentNo	繳費代碼	String(14)	
	ExpireDate	繳費期限	Int	
	PaymentURL	繳費連結	String(100)	由綠界科技提供手機上顯示的三段式繳費條碼網頁
BarcodeInfo		超商條碼資訊	JSON	
	ExpireDate	繳費期限	String(20)	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
ATMInfo		ATM 資訊	JSON	
	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
CustomField		特店自訂欄位	String(200)	提供特店使用客制化欄位

Data 解密後：

```
{
  "RtnCode": 1,
  "RtnMsg": "Success",
  "MerchantID": "3002607",
  "OrderInfo": {
    "MerchantTradeNo": "20180914001",
    "TradeNo": "1809261503338172",
    "PaymentDate": "2018/09/26 14:59:54",
    "TradeAmt": 500,
    "PaymentType": "Credit",
    "TradeDate": "2018/09/26 14:59:54",
    "ChargeFee": 10,
    "TradeStatus": "1"
  },
  "CVSInfo": {
    "PaymentNo": "12345678901234",
    "ExpireDate": 7,
    "PaymentURL": "https://aa.aa.aa"
  },
  "BarcodeInfo": {
    "ExpireDate": "2020/12/31 15:12:20",
    "Barcode1": "123456789",
    "Barcode2": "1234567890123456",
    "Barcode3": "123456789012345"
  },
  "ATMInfo": {
    "BankCode": "800",
    "vAccount": "1234567890",
    "ExpireDate": "2020/12/31"
  },
}
```

10. 綁定信用卡流程說明

綠界特店會員信用卡綁定服務對於有綁定信用卡需求的特約商店，提供綁定信用卡介接技術與幕後交易的授權方式，並且提供刪除綁定的信用卡與查詢綁定資料之服務；綁定信用卡畫面為嵌入式，特店可將綠界所產生的綁定信用卡畫面嵌入至自己的網頁，並可自行修改畫面，以達到自行設計風格樣式的需求。



處理角色	流程名稱	處理說明
特店 Web	1.初始化 SDK	開始使用 SDK 需先初始化，並帶入環境變數
綠界 JS	2.建立 SDK 初始化資料	SDK 建立初始化資料
特店 Server	3.呼叫取得 廠商驗證碼 API	特店確定訂單內容後，傳送訂單資料到綠界並取得廠商驗證碼 (Token)。API 詳細說明請參考 取得廠商驗證碼(綁定信用卡功能) 章節
綠界 Server	4.建立廠商驗證碼	綠界 Server 接收到綁卡訂單資料後，產生一組廠商驗證碼(Token)並回傳至特店 Server
特店 Server	5.取得廠商驗證碼	接收綠界回傳的廠商驗證碼(Token)
特店 Web	6. 取得綁定信用卡功能	呼叫綠界提供之 JS SDK 中的 addBindingCard() 並帶入廠商驗證碼 (Token)，顯示綁定信用卡功能畫面
綠界 JS	7.產生綁定信用卡畫面	綠界 JS SDK 透過特店帶入的 Token，產生綁定信用卡功能畫面。特店可將此畫面內嵌至特店欲放置的 Web 畫面， 且必須自行實作綁卡確認按鈕
特店 Web	8.取得綁卡代碼 BindCardPayToken	當使用者按下特店自行實作的確認按鈕時，特店必須呼叫綠界 JS SDK 的 getBindCardPayToken()
綠界 JS	9.建立 BindCardPayToken	建立 BindCardPayToken 資料
特店 Web	10.接收 BindCardPayToken	特店接收 BindCardPayToken 並將 BindCardPayToken 傳至特店 Server
特店 Server	11. 建立綁卡交易	呼叫建立綁定信用卡交易 API，並帶入 BindCardPayToken
綠界 Server	12.綁卡交易授權 (含 3D 驗證)	將綁卡資料送至綠界 Server 建立綁卡交易並將信用卡資料送至銀行進行授權與 3D 驗證
特店 Server	13.取得 3D 驗證 URL	取得 3D 驗證 URL 並將 URL 傳送至特店 Web
特店 Web	14.開啟 3D 驗證 URL	開啟 3D 驗證 URL
銀行	15.進行 3D 驗證	銀行進行 3D 驗證
綠界 Server	16.通知銀行 3D 驗證結果	接收 3D 驗證結果
特店 Web	17.接收付款結果	透過 OrderResultURL 接收綁卡結果
特店 Server	18.接收付款結果	透過 ReturnURL 幕後接收綁卡結果通知，接收綁定信用卡代碼 BindCardID。
特店 Server	19.放棄綁卡交易授權	因建立綁卡交易必須帶入特店的最低交易金額，此功能會實際送至銀行進行授權。特店務必在綁卡完成後，操作放棄綁卡交易授權，以避免向使用者收取費用
綠界 Server	20.放棄綁卡交易授權	綠界取消該筆綁卡資料的信用卡授權
特店 Server	21.成功取消綁卡授權金額	成功取消綁卡授權金額，綁卡流程結束

11. 綁定信用卡使用情境說明

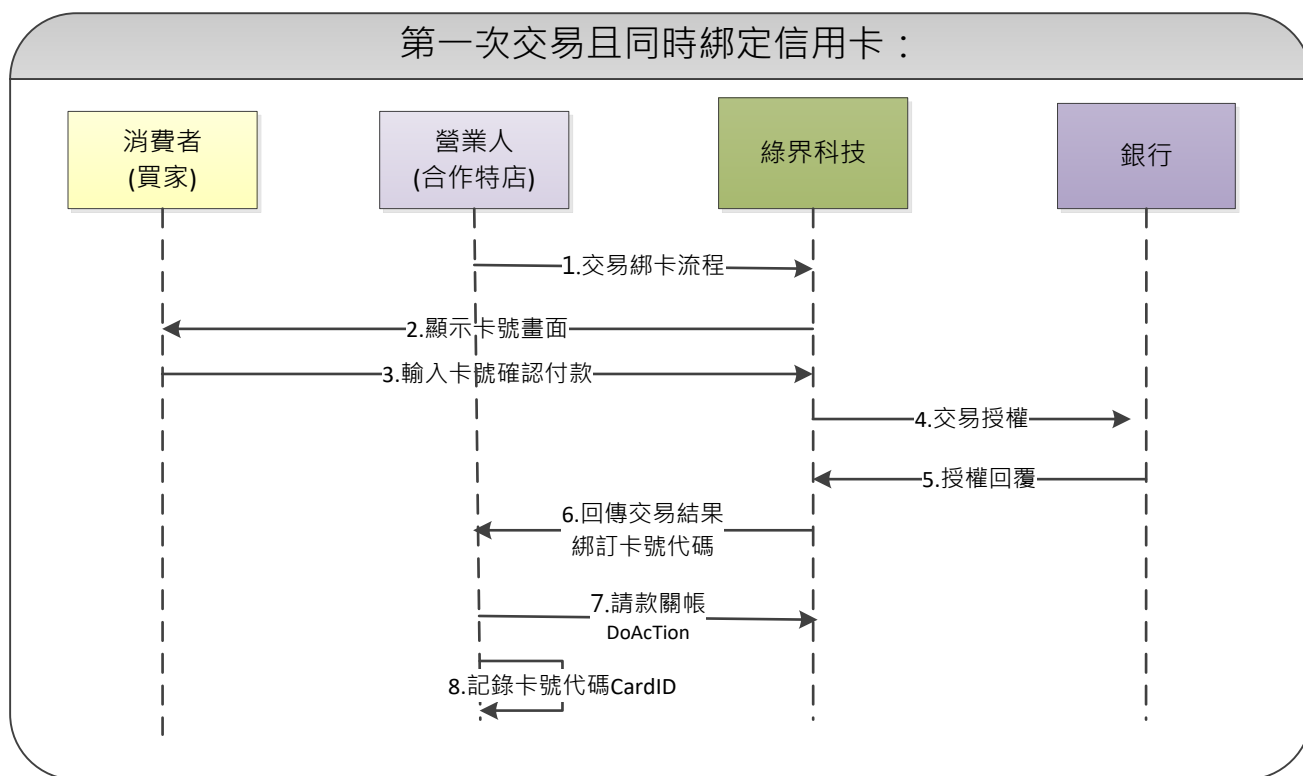
綁定信用卡功能適用於不定期不定額扣款，於特店自主發動扣款，第一次結帳買家需要輸入信用卡資訊並完成 3D 驗證，第二次扣款時機與結帳畫面皆可由特店自行設計。

※注意事項:

第一次綁定信用卡時會進行 3D 驗證，但如消費者端之信用卡不支援 3D 功能，依然可綁卡成功。如廠商不支援此種信用卡進行綁定，可向綠界業務單位申請絕對 3D 功能。

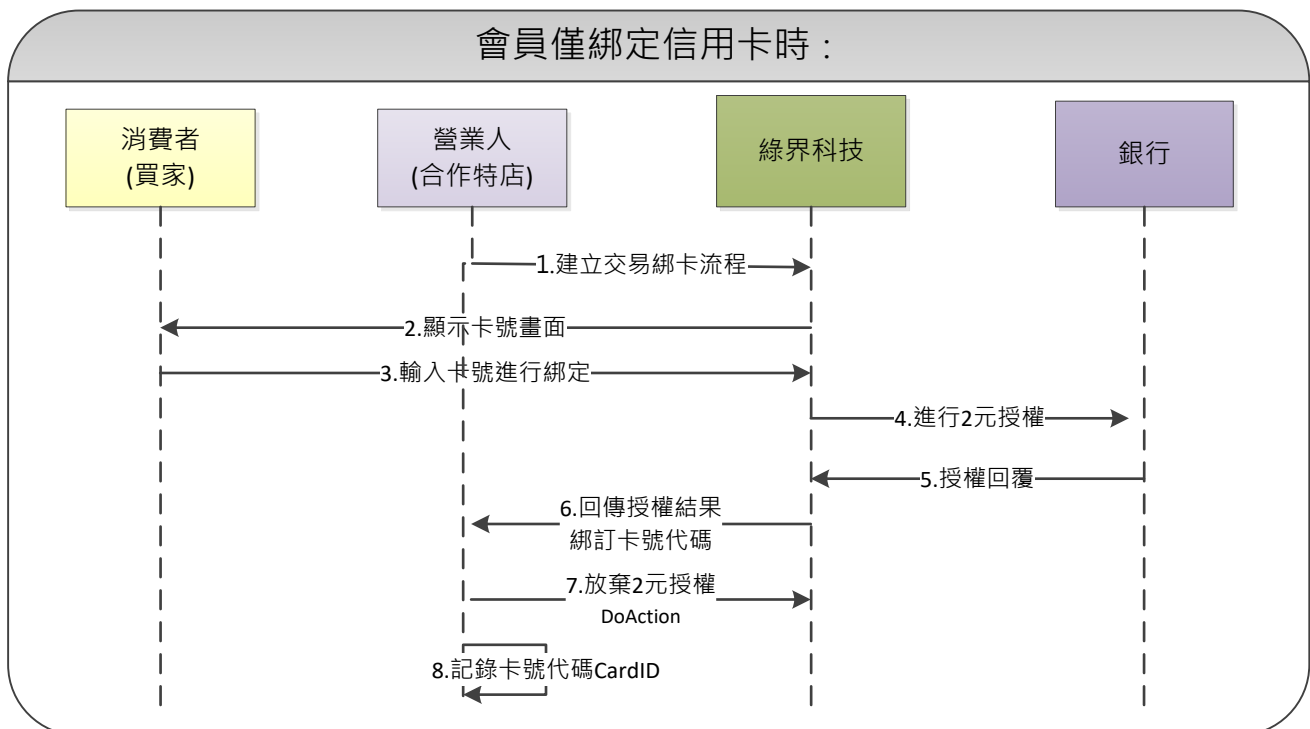
情境一：第一次交易且同時綁定信用卡：

1. 特店進行[[交易綁卡流程](#)]，進行交易結帳
2. 綠界科技顯示卡號交易畫面
3. 消費者輸入卡號進行付款
4. 綠界科技送交易授權到銀行進行驗證
5. 銀行回覆授權結果
6. 綠界回覆特店交易授權結果及綁定卡號代碼
7. 特店呼叫[[信用卡請退款功能](#)] 進行請款關帳
8. 特店依會員編號記錄綁定卡號代碼



情境二：會員僅綁定信用卡：

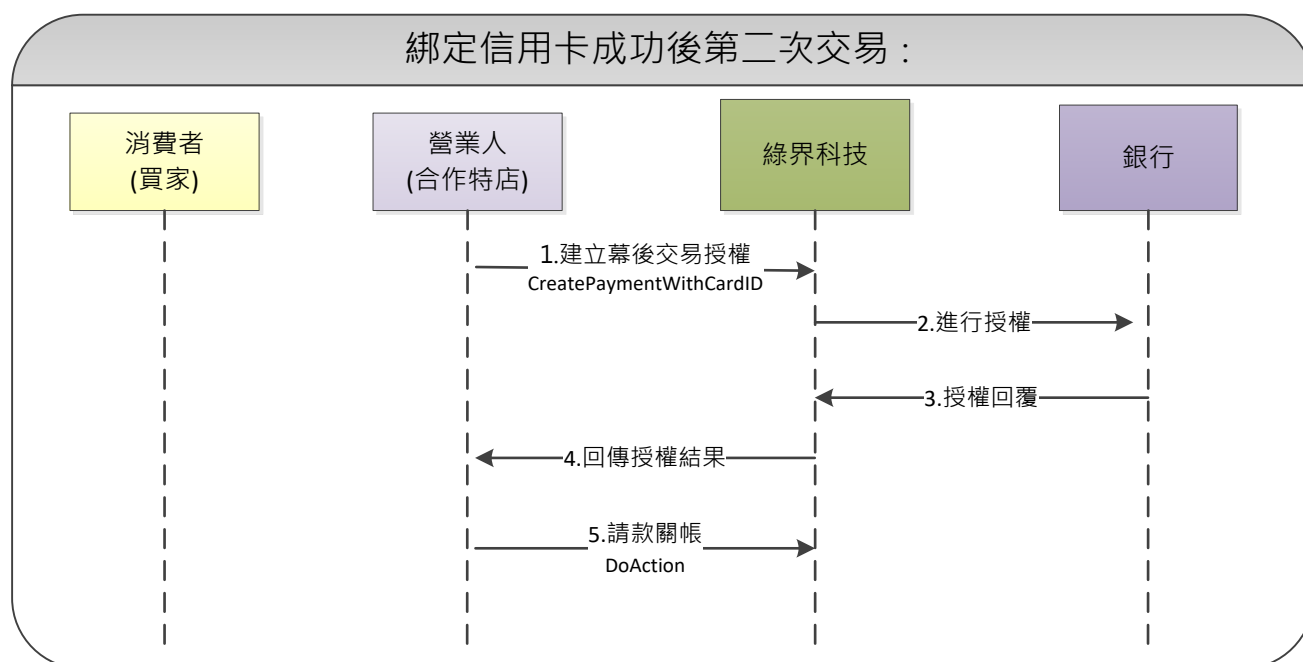
1. 特店進行[[交易綁卡流程](#)]，進行 2 元授權
2. 綠界科技顯示卡號交易畫面
3. 消費者輸入卡號進行付款
4. 綠界科技送交易授權到銀行進行驗證
5. 銀行回覆授權結果
6. 綠界回覆特店交易授權結果及綁定卡號
7. 特店呼叫[[信用卡請退款功能](#)] 進行放棄 2 元授權
8. 特店依會員編號記錄綁定卡號



情境三：綁定信用卡成功後第二次交易：

1. 特店呼叫[[幕後交易授權](#)]，進行交易授權
2. 綠界科技送交易授權到銀行進行驗證
3. 銀行回覆授權結果
4. 綠界回覆特店交易授權結果
5. 特店呼叫[[信用卡請退款功能](#)] 進行交易請款

綁定信用卡成功後第二次交易：



12.取得廠商驗證碼(綁定信用卡功能/Server)

- 在使用綠界特店會員信用卡綁定服務之前，特店 Server 必須先向綠界 Server 取得一組廠商驗證碼(Token)。特店 Server 得到驗證碼後，必須將驗證碼傳給特店 Web 做產生綁定信用卡畫面功能之用。

※注意事項:

一個特店會員最多只能綁定五張信用卡。

- 介接路徑
 - 正式環境：https://ecpg.ecpay.com.tw/Merchant/GetTokenbyBindingCard
 - 測試環境：https://ecpg-stage.ecpay.com.tw/Merchant/GetTokenbyBindingCard
- 特店傳入參數說明：(JSON 格式，名稱若有紅色星號*為必填欄位)
 - Content Type : application/json
 - HTTP Method : POST

參數	參數名稱	型態	說明
*MerchantID	特店編號	String(10)	特店編號 若為平台商時，參數請帶平台商所綁的特店編號
*RqHeader	傳輸資料	JSON	
	*Timestamp	Number	請將傳輸時間轉換為時間戳(GMT+8)，綠界會利用此參數將當下的時間轉為 Unix TimeStamp 來驗證此次介接的時間區間。 注意事項: 若時間戳跟綠界伺服器接收到時間超過 10 分鐘時，交易會失敗無法進行
	*Revision	String(10)	串接文件版號。Ex: 1.0.0
*Data	加密資料	String	加密過 JSON 格式的資料。 加密方法說明

範例:

```
{
  "MerchantID": "3002607",
  "RqHeader": {
    "Timestamp": 1234567890,
    "Revision": "1.0.0"
  },
  "Data": "..."
}
```

}

- Data 參數說明：請在加密前對參數值進行 URLEncode

參數	參數名稱	型態	說明
PlatformID	特約合作平台商代號	String(10)	為專案合作的平台商使用。一般特店或平台商本身介接，則參數請帶空值。若為專案合作平台商的特店使用時，則參數請帶平台商所綁的特店編號 [MerchantID]。
*GatewayMerchantID	閘道商特店編號	String(10)	實際建立訂單的金流模式，會依照綠界廠商後台的閘道介接設定內容為主。
*ConsumerInfo	特店會員資訊	JSON	
*MerchantMemberID	特店會員編號	String(60)	MerchantMemberID 為特店的會員系統中的會員編號(會員 ID)，而一個特店會員最多只能綁定五張信用卡。
Email	信用卡持卡人電子信箱	String(200)	
Phone	信用卡持卡人電話	String(60)	可接受帶國碼的形式，但不可帶+號 ex: 886912345678
Name	信用卡持卡人姓名	String(50)	※注意事項： 該欄位可接受中文、英文與部分特殊符號。 可支援的特殊符號為: ,.()/-
CountryCode	國別碼	String(3)	持卡人帳單地址國別碼，請參考 ISO3166 。臺灣請填寫 158
Address	地址	String(50)	持卡人帳單地址
*OrderInfo	綁卡訂單資訊	JSON	
*MerchantTradeDate	特店交易時間	String(20)	格式為 yyyy/MM/dd HH:mm:ss
*MerchantTradeNo	特店交易編號	String(20)	為唯一值，不可重複使用。英數字大小寫混和
*TotalAmount	綁卡交易授權金額	Int	綁定信用卡前必須輸入該特店的最低交易金額，作為綁定信用卡 3D 驗證使用。 最低交易金額可於綠界廠商管理後台>>廠商專區>>金物流、電子發票合約及費率中查詢。 ※注意事項： 此金額為實際的交易授權金額，特店必須於綁定信用卡成功後將此交易取消授權。
*TradeDesc	綁卡交易描述	String(200)	請自行帶入綁卡交易的描述
*ItemName	交易名稱	String(400)	
*ReturnURL	綁卡回傳結果	String(200)	當消費者綁卡完成後，綠界會將綁卡結果參數以幕後方式回傳到該網址。
ConsumerInfo	特店會員資訊	JSON	當特店擁有自己的會員系統時，可將會員資料帶入

MerchantMemberID	特店會員編號	String(60)	當[RememberCard] = 1，此欄位必填 請帶入特店的會員編號
Email	信用卡持卡人 電子信箱	String(30)	
Phone	信用卡持卡人 電話	String(60)	注意事項： 可接受帶國碼的形式，但不可帶+號 ex: 886912345678
Name	信用卡持卡人 姓名	String(50)	注意事項： 該欄位可接受中文、英文與部分特殊符號。 可支援的特殊符號為: ,.()/-
CountryCode	國別碼	String(3)	持卡人帳單地址國別碼，請參考 ISO3166 。臺灣請填寫 158
Address	地址	String(50)	持卡人帳單地址
*OrderResultURL	3D 驗證綁卡回 傳結果	String(200)	綁定信用卡需使用 3D 驗證。當 3D 驗證完成後，綠界會將綁卡結果參數以幕前方式回傳到該網址。
CustomField	自訂欄位	String(200)	提供特店使用客制化欄位 請勿傳入超過長度 200 的內容，避免該欄位資訊被截斷

Data 範例：

```
{
  "GatewayMerchantID": "3002607",
  "ConsumerInfo": {
    "MerchantMemberID": "test123456"
  },
  "OrderInfo": {
    "MerchantTradeDate": "2022/05/23 14:10:00",
    "MerchantTradeNo": "1234567890",
    "TotalAmount": 500,
    "TradeDesc": "綁定信用卡",
    "ItemName": "V4 綁定信用卡",
    "ReturnURL": "https://aa.aa.aa"
  },
  "OrderResultURL": "https://bb.bb.bb"
}
```

- 綠界回傳參數說明：(JSON 格式)

參數	參數名稱	型態	說明
MerchantID	特店編號	String(10)	特店編號
RpHeader	回傳資料	JSON	
Timestamp	回傳時間	Number	時間戳 Unix timestamp
TransCode	回傳代碼	Int	1 代表 API 傳輸資料(MerchantID, RqHeader, Data)接收成功，實際的 API 執行結果狀態請參考 RtnCode
TransMsg	回傳訊息	String(200)	回傳訊息
Data	加密資料	String	加密過 JSON 格式的資料。 加密方法說明

範例:

```
{
  "MerchantID": "3002607",
  "RpHeader": {
    "Timestamp": 1234564848
  },
  "TransCode": 1,
  "TransMsg": "Success",
  "Data": "..."
}
```

- Data 參數說明：請在解密後對參數值進行 URLDecode

參數	參數名稱	型態	說明
RtnCode	交易狀態	Int	1 代表 API 執行成功，代表取得廠商驗證碼成功。其餘代碼均為失敗。失敗代碼請參考 交易訊息代碼表
RtnMsg	回應訊息	String(200)	
RtnCode 為 1 時，以下參數才會有值：			
PlatformID	平台商編號	String(10)	特約合作平台商特店編號。
MerchantID	特店編號	String(10)	
Token	交易代碼	String(64)	可使用的效期為 30 分鐘
TokenExpireDate	交易代碼到期時間	String(20)	格式為 yyyy/MM/dd HH:mm:ss

Data 解密後:

```
{
  "RtnCode": 1,
  "RtnMsg": "Success",
  "PlatformID": "1234567890",
  "MerchantID": "1234567890",
  "Token": "m12dae4846446sq",
  "TokenExpireDate": "2020/09/18 15:39:10"
}
```

13. 綁定信用卡嵌入式畫面 SDK(JS)

- 應用場景:

適用於前端 Web，提供特店綁定信用卡功能界面，當特店取得廠商驗證碼(Token)後，帶入 SDK 以取得綁定信用卡功能。

- 使用方式:

1. 引用外部 JS 元件:

```
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
```

2. 引用綠界 JS 元件:

- 正式環境：

```
<script src="https://ecpg.ecpay.com.tw/Scripts/sdk-1.0.0.js?t=20210121100116"></script>
```

- 測試環境：

```
<script src="https://ecpg-stage.ecpay.com.tw/Scripts/sdk-1.0.0.js?t=20210121100116"></script>
```

※注意事項:

(1) 引用綠界 JS 元件時，必須放在<div id="ECPayPayment"></div>的後面

3. Html 區塊定義:

```
<div id="ECPayPayment"></div> //呈現付款方式畫面
```

※注意事項:

(2) 定義 CSS 參數時，請勿與站內付 2.0 SDK 定義的相同，請[參考 CSS 參數定義列表](#)。

4. **Initialize:** 初始化 SDK，環境參數請參考下段說明。

```
ECPay.initialize( ServerType, IsLoading, callBack(errMsg) )
```

■ 特店帶入參數:

參數	參數名稱	型態	說明
*ServerType	環境參數	String	Stage:測試環境 ECPay.ServerType.Stage Prod:正式環境 ECPay.ServerType.Prod
IsLoading	是否顯示等待畫面	Int	0:不顯示等待畫面 1:顯示等待畫面 如無設定該值，預設為 1
*callback	CallBack	Function	執行回呼函式 1.errMsg - 執行失敗時，可在 callback 接收錯誤訊息 2.需於此處呼叫 ECPay.createPayment

5. 取得綁定信用卡功能

ECPay.addBindingCard (Token, Language, callback(errMsg))

■ 帶入參數:

參數	參數名稱	型態	說明
*Token	廠商驗證碼	String	初始化綁定信用卡畫面代碼，此參數具 30 分鐘時效性
*Language	多語系	String	可使用以下方式設定使用語系 繁中: ECPay.Language.zhTW 英語: ECPay.Language.enUS
*callback	Callback	Function	errMsg - 執行失敗時，可在 callback 接收錯誤訊息，成功時該參數為 null

6. 取得綁定信用卡代碼(BindCardPayToken)

ECPay.getBindCardPayToken(callback(BindCardPayToken, errMsg))

■ 回傳參數:

參數	參數名稱	型態	說明
BindCardPayToken	綁定信用卡代碼	String	
errMsg	錯誤訊息	String	errMsg - 執行失敗時，可在 callback 接收錯誤訊息

7. GetLanguage: 取得設定語系

ECPay.getLanguage()

■ 綠界回傳參數:

內容	說明
zh-TW · en-US	切換語系時，可取出於 SDK 目前設定的語系(預設:zh-TW)

● 範例程式:

```
//初始化 SDK
ECPay.initialize(serverType, function (errMsg) {
    var lang = getCookie("language");
    if (!lang) {
        lang = ECPay.Language.zhTW;
    }
    ECPay.addBindingCard("55a026b8df314f9bb2dc808a5fe21c9d", lang, function (errMsg) {
    });
    if ($(this).val() === ECPay.getLanguage()) {
        $(this).get(0).selected = true;
        return false;
    };
    ECPay.getBindCardPayToken( function (paymentInfo, errMsg) {
        console.log(paymentInfo)
        console.log(errMsg)
        if (errMsg != null)
        {
            ErrHandle(errMsg);
            return;
        };
    });
});
```


14. 建立綁定信用卡訂單(Server)

- 應用場景:

特店 Server 需將從特店 Web 取得到的 BindCardPayToken，送至綠界 Server 建立綁定信用卡交易。

- 介接路徑：

- 正式環境：https://ecpg.ecpay.com.tw/Merchant/CreateBindCard
- 測試環境：https://ecpg-stage.ecpay.com.tw/Merchant/CreateBindCard

- 特店傳入參數說明：(JSON 格式)

- Content Type : application/json
- HTTP Method : POST

參數	參數名稱	型態	說明
*MerchantID	特店編號	String(10)	特店編號 若為平台商時，參數請帶平台商所綁的特店編號
*RqHeader	傳輸資料	JSON	
	*Timestamp	傳輸時間	Number 請將傳輸時間轉換為時間戳(GMT+8)，綠界會利用此參數將當下的時間轉為 Unix TimeStamp 來驗證此次介接的時間區間。 注意事項： 若時間戳跟綠界伺服器接收到時間超過 10 分鐘時，交易會失敗無法進行
	*Revision	串接版號	String(10) 請參考文件封面的 文件版號 。Ex: 1.0.0
*Data	加密資料	String	加密過 JSON 格式的資料。 加密方法說明

範例:

```
{
  "MerchantID": "3002607",
  "RqHeader": {
    "Timestamp": 1234567890,
    "Revision": "1.0.0"
  },
  "Data": "..."
}
```

- Data 參數說明：請在加密前對參數值進行 URLEncode

參數	參數名稱	型態	說明
PlatformID	平台商代號	String(10)	平台商特店編號
*GatewayMerchantID	閘道商特店編號	String(10)	當特店身為閘道商時，帶入此參數才能使用綠界廠商後台的閘道介接設定內容功能。實際建立訂單的金流模式，會依照設定內容為主。
*BindCardPayToken	綁定信用卡代碼	String(64)	
*MerchantMemberID	特店會員編號	String(60)	

Data 範例：

```
{
  "MerchantID": "3002607",
  "BindCardPayToken": "enter your token",
  "MerchantMemberID": "test123456789"
}
```

- 綠界回傳參數說明：(JSON 格式)

參數	參數名稱	型態	說明
MerchantID	特店編號	String(10)	
RpHeader	回傳資料	JSON	
Timestamp	回傳時間	Number	時間戳 Unix timestamp
TransCode	回傳代碼	Int	1 代表 API 傳輸資料(MerchantID, RpHeader, Data)接收成功，實際的 API 執行結果狀態請參考 RtnCode
TransMsg	回傳訊息	String(200)	回傳訊息
Data	加密資料	String	加密過 JSON 格式的資料。 加密方法說明

範例:

```
{
  "MerchantID": "3002607",
  "RpHeader": {
    "Timestamp": 1234567890
  },
  "TransCode": 1,
  "TransMsg": "成功",
  "Data": "..."
}
```

}

- Data 參數說明：請在解密後對參數值進行 URLDecode

參數	參數名稱	型態	說明
RtnCode	交易狀態	Int	1 代表 API 執行綁定信用卡交易成功，其餘代碼均為失敗。失敗代碼請參考 交易訊息代碼表
RtnMsg	回應訊息	String(200)	
RtnCode 為 1 時，以下參數才會有值：			
PlatformID	平台商代號	String(10)	平台商特店編號
MerchantID	特店編號	String(10)	注意事項： 當建立綁卡訂單且帶入的參數為 GatewayMerchantID 時，實際回傳的特店編號會依據廠商後台設定的結果為準。
OrderInfo	訂單資訊	JSON	
MerchantTradeNo	特店交易編號	String(20)	不可重複
ThreeDInfo	3D 驗證資訊	JSON	
ThreeDURL	3D 驗證 URL	String(200)	3D 驗證連結，請勿使用 iframe 方式開啟。

Data 範例：

```
{
  "RtnCode": 1,
  "RtnMsg": "Success",
  "MerchantID": "2000132",
  "OrderInfo": {
    "MerchantTradeNo": 1234567890
  },
  "ThreeDInfo": {
    "ThreeDURL": "https://3durl.com.tw"
  }
}
```

15.查詢會員綁定信用卡資料(Server)

- 應用場景:

提供特店查詢綁定信用卡資料，可透過此 API 來查詢會員的綁卡清單。

- 介接路徑：

- 正式環境：<https://ecpg.ecpay.com.tw/Merchant/GetMemberBindCard>
- 測試環境：<https://ecpg-stage.ecpay.com.tw/Merchant/GetMemberBindCard>

- 特店傳入參數說明：(JSON 格式)

- Content Type : application/json
- HTTP Method : POST

參數	參數名稱	型態	說明
*MerchantID	特店編號	String(10)	特店編號 若為平台商時，參數請帶平台商所綁的特店編號
*RqHeader	傳輸資料	JSON	
	*Timestamp	傳輸時間	Number 請將傳輸時間轉換為時間戳(GMT+8)，綠界會利用此參數將當下的時間轉為 Unix TimeStamp 來驗證此次介接的時間區間。 注意事項: 若時間戳跟綠界伺服器接收到時間超過 10 分鐘時，交易會失敗無法進行
	*Revision	串接版號	String(10) 請參考文件封面的 文件版號 。Ex: 1.0.0
*Data	加密資料	String	加密過 JSON 格式的資料。 加密方法說明

範例:

```
{
  "MerchantID": "3002607",
  "RqHeader": {
    "Timestamp": 1234567890,
    "Revision": "1.0.0"
  },
  "Data": "..."
}
```

- Data 參數說明：請在加密前對參數值進行 URLEncode

參數	參數名稱	型態	說明
PlatformID	平台商代號	String(10)	
*MerchantID	特店編號	String(10)	若為閘道商時，參數請帶閘道商的特店編號[MerchantID]
*MerchantMemberID	特店會員編號	String(60)	
MerchantTradeNo	特店交易編號	String(20)	

Data 範例：

```
{
  "MerchantID": "3002607",
  "MerchantMemberID": "test012345",
  "MerchantTradeNo": "20180914001"
}
```

- 綠界回傳參數說明：(JSON 格式)

參數	參數名稱	型態	說明
MerchantID	特店編號	String(10)	
RpHeader	回傳資料	JSON	
Timestamp	回傳時間	Number	時間戳 Unix timestamp
TransCode	回傳代碼	Int	1 代表 API 傳輸資料(MerchantID, RqHeader, Data)接收成功，實際的 API 執行結果狀態請參考 RtnCode
TransMsg	回傳訊息	String(200)	回傳訊息
Data	加密資料	String	加密過 JSON 格式的資料。 加密方法說明

範例:

```
{
  "MerchantID": "3002607",
  "RpHeader": {
    "Timestamp": 1234567890
  },
  "TransCode": 1,
  "TransMsg": "成功",
  "Data": "..."
}
```

● Data 參數說明：請在解密後對參數值進行 URLDecode

參數	參數名稱	型態	說明
RtnCode	交易狀態	Int	1 代表 API 執行成功，其餘代碼均為失敗。 失敗代碼請參考 交易訊息代碼表
RtnMsg	回應訊息	String(200)	
RtnCode 為 1 時，以下參數才會有值：			
PlatformID	平台商代號	String(10)	平台商特店編號
MerchantID	特店編號	String(10)	
MerchantMemberID	特店會員編號	String(60)	
BindCardList	綁定信用卡清單	Array	
	Card6No	信用卡卡號前六碼	String(6)
	Card4No	信用卡卡號前四碼	String(4)
	CardValidYY	信用卡有效年	String(2) 格式為 YY
	CardValidMM	信用卡有效月	String(2) 格式為 MM
	BindCardID	綁定信用卡代碼	String(64)

Data 範例：

```
{
  "RtnCode": 1,
  "RtnMsg": "Success",
  "MerchantID": "3002607",
  "BindCardList": [
    {
      "Card6No": "123456",
      "Card4No": "4123",
      "CardValidYY": "25",
      "CardValidMM": "11",
      "BindCardID": "ab123456acacssc",
    },
    {
      "Card6No": "654123",
      "Card4No": "7894",
      "CardValidYY": "27",
      "CardValidMM": "12",
      "BindCardID": "ab789456acac",
    }
  ]
}
```

```
}  
]  
}
```

16. 刪除綁定信用卡(Server)

- 應用場景:

提供特店刪除綁定信用卡功能，可透過此 API 來指定刪除某張已綁定的信用卡。

- 介接路徑：

- 正式環境：https://ecpg.ecpay.com.tw/Merchant/DeleteMemberBindCard
- 測試環境：https://ecpg-stage.ecpay.com.tw/Merchant/DeleteMemberBindCard

- 特店傳入參數說明：(JSON 格式)

- Content Type : application/json
- HTTP Method : POST

參數	參數名稱	型態	說明
*MerchantID	特店編號	String(10)	特店編號 若為平台商時，參數請帶平台商所綁的特店編號
*RqHeader	傳輸資料	JSON	
*Timestamp	傳輸時間	Number	請將傳輸時間轉換為時間戳(GMT+8)，綠界會利用此參數將當下的時間轉為 Unix TimeStamp 來驗證此次介接的時間區間。 注意事項： 若時間戳跟綠界伺服器接收到時間超過 10 分鐘時，交易會失敗無法進行
*Revision	串接版號	String(10)	請參考文件封面的 文件版號 。Ex: 1.0.0
*Data	加密資料	String	加密過 JSON 格式的資料。 加密方法說明

範例:

```
{
  "MerchantID": "3002607",
  "RqHeader": {
    "Timestamp": 1234567890,
    "Revision": "1.0.0"
  },
  "Data": "..."
}
```


- Data 參數說明：請在加密前對參數值進行 URLEncode

參數	參數名稱	型態	說明
PlatformID	平台商代號	String(10)	
*MerchantID	特店編號	String(10)	若為閘道商時，參數請帶閘道商的特店編號[MerchantID]
*BindCardID	綁定信用卡代碼	String(64)	

Data 範例：

```
{
  "MerchantID": "3002607",
  "BindCardID": "abc1234456abc"
}
```

- 綠界回傳參數說明：(JSON 格式)

參數	參數名稱	型態	說明
MerchantID	特店編號	String(10)	
RpHeader	回傳資料	JSON	
Timestamp	回傳時間	Number	時間戳 Unix timestamp
TransCode	回傳代碼	Int	1 代表 API 傳輸資料(MerchantID, RqHeader, Data)接收成功，實際的 API 執行結果狀態請參考 RtnCode
TransMsg	回傳訊息	String(200)	回傳訊息
Data	加密資料	String	加密過 JSON 格式的資料。 加密方法說明

範例:

```
{
  "MerchantID": "3002607",
  "RpHeader": {
    "Timestamp": 1234567890
  },
  "TransCode": 1,
  "TransMsg": "成功",
  "Data": "..."
}
```

- Data 參數說明：請在解密後對參數值進行 URLDecode

參數	參數名稱	型態	說明
RtnCode	交易狀態	Int	1 代表 API 執行成功，其餘代碼均為失敗。 失敗代碼請參考 交易訊息代碼表
RtnMsg	回應訊息	String(200)	

Data 範例：

```
{  
  "RtnCode": 1,  
  "RtnMsg": "Success"  
}
```

17. 幕後交易授權(Server)

- 應用場景:

提供特店利用綁定信用卡取得的 BindCardID 進行幕後交易授權。

- 介接路徑：

- 正式環境：https://ecpg.ecpay.com.tw/Merchant/CreatePaymentWithCardID
- 測試環境：https://ecpg-stage.ecpay.com.tw/Merchant/CreatePaymentWithCardID

- 特店傳入參數說明：(JSON 格式)

- Content Type : application/json
- HTTP Method : POST

參數	參數名稱	型態	說明
*MerchantID	特店編號	String(10)	特店編號 若為平台商時，參數請帶平台商所綁的特店編號
*RqHeader	傳輸資料	JSON	
*Timestamp	傳輸時間	Number	請將傳輸時間轉換為時間戳(GMT+8)，綠界會利用此參數將當下的時間轉為 Unix TimeStamp 來驗證此次介接的時間區間。 注意事項： 若時間戳跟綠界伺服器接收到時間超過 10 分鐘時，交易會失敗無法進行
*Revision	串接版號	String(10)	請參考文件封面的 文件版號 。Ex: 1.0.0
*Data	加密資料	String	加密過 JSON 格式的資料。 加密方法說明

範例:

```
{
  "MerchantID": "3002607",
  "RqHeader": {
    "Timestamp": 1234567890,
    "Revision": "1.0.0"
  },
  "Data": "..."
}
```

● Data 參數說明：請在加密前對參數值進行 URLEncode

參數	參數名稱	型態	說明
PlatformID	平台商代號	String(10)	平台商特店編號
*GatewayMerchantID	閘道商特店編號	String(10)	當特店身為閘道商時，帶入此參數才能使用綠界廠商後台的閘道介接設定內容功能。實際建立訂單的金流模式，會依照設定內容為主。
*BindCardID	綁定信用卡代碼	String(64)	
*OrderInfo	交易訂單資訊	JSON	
*MerchantTradeDate	特店交易時間	String(20)	格式為 yyyy/MM/dd HH:mm:ss
*MerchantTradeNo	特店交易編號	String(20)	
*TotalAmount	交易金額	Int	※注意事項： 交易金額不可超過 20 萬新台幣
*ReturnURL	付款回傳結果	String(200)	
*TradeDesc	交易描述	String(200)	
*ItemName	商品名稱	String(400)	
*ConsumerInfo	特店會員資訊	JSON	
*MerchantMemberID	特店會員編號	String(60)	
Email	信用卡持卡人電子信箱	String(200)	
Phone	信用卡持卡人電話	String(60)	
Name	信用卡持卡人姓名	String(50)	
CountryCode	國別碼	String(3)	
Address	地址	String(50)	
CustomField	自訂欄位	String(200)	請勿傳入超過長度 200 的內容，避免該欄位資訊被截斷

Data 範例：

```
{
  "MerchantID": "3002607",
  "BindCardID": "abc123456ac",
  "OrderInfo": {
    "MerchantTradeDate": "2022/05/23 14:10:00",
    "MerchantTradeNo": "1234567890",
    "TotalAmount": 500,
```

```

    "TradeDesc": "交易描述",
    "ItemName": "泡麵#清潔用品",
    "ReturnURL": "https://aa.aa.aa"
  },
  "ConsumerInfo": {
    "MerchantMemberID": "test123456"
  }
}

```

● 綠界回傳參數說明：(JSON 格式)

參數	參數名稱	型態	說明
MerchantID	特店編號	String(10)	
RpHeader	回傳資料	JSON	
Timestamp	回傳時間	Number	時間戳 Unix timestamp
TransCode	回傳代碼	Int	1 代表 API 傳輸資料(MerchantID, RqHeader, Data)接收成功，實際的 API 執行結果狀態請參考 RtnCode
TransMsg	回傳訊息	String(200)	回傳訊息
Data	加密資料	String	加密過 JSON 格式的資料。 加密方法說明

範例:

```

{
  "MerchantID": "3002607",
  "RpHeader": {
    "Timestamp": 1234567890
  },
  "TransCode": 1,
  "TransMsg": "成功",
  "Data": "..."
}

```

● Data 參數說明：請在解密後對參數值進行 URLDecode

參數	參數名稱	型態	說明
RtnCode	交易狀態	Int	1 代表 API 執行成功，其餘代碼均為失敗。 失敗代碼請參考 交易訊息代碼表
RtnMsg	回應訊息	String(200)	
RtnCode 為 1 時，以下參數才有值：			
PlatformID	平台商代號	String(10)	平台商特店編號
MerchantID	特店編號	String(10)	注意事項： 當建立幕後交易授權訂單且帶入的參數為 GatewayMerchantID 時，實際回傳的特店編號會依據廠商後台設定的結果為準。
OrderInfo	訂單資訊	JSON	
MerchantTradeNo	特店交易編號	String(20)	
TradeNo	綠界交易編號	String(20)	請保存綠界的交易編號與特店交易編號 [MerchantTradeNo] 的關聯。
TradeAmt	交易金額	Int	
TradeDate	訂單成立時間	String(20)	格式為 yyyy/MM/dd HH:mm:ss
PaymentType	付款方式	String(20)	
PaymentDate	付款時間	String(20)	格式為 yyyy/MM/dd HH:mm:ss
ChargeFee	手續費	Number	
TradeStatus	交易狀態	String(8)	回傳值： 若為 0 時，代表交易訂單付款失敗 若為 1 時，代表交易訂單付款成功
CardInfo	授權資訊	JSON	
AuthCode	銀行授權碼	String(6)	
Gwsr	授權交易單號	Int	
ProcessDate	交易時間	String(20)	yyyy/MM/dd HH:mm:ss
Amount	金額	Int	
Card6No	信用卡卡號前六碼	String(6)	
Card4No	信用卡卡號末四碼	String(4)	
CoBrandingInfo	聯名卡資料	Array	
CoBrandingCode	聯名卡代號	String(20)	
Comment	備註	String(100)	
CustomField	特店自訂欄位	String(200)	

Data 範例：

```
{
  "RtnCode": 1,
  "RtnMsg": "Success",
  "PlatformID": "1234567890",
  "MerchantID": "1234567890",
  "CardInfo": {
    "Card6No": "123456",
    "Card4No": "1234",
    "IssuingBank": "中國信託商業銀行",
    "IssuingBankCode": "822",
    "AuthCode": "654123",
    "Gwsr": 125,
    "ProcessDate": "2022/09/18 15:39:10",
    "Amount": 2,
    "Eci": 1,
  },
  "OrderInfo": {
    "MerchantTradeNo": "test123466",
    "TradeNo": "123456789",
    "PaymentDate": "2022/09/18 15:39:10",
    "TradeAmt": 2,
    "PaymentType": "Credit",
    "TradeDate": "2022/09/18 15:39:10",
    "ChargeFee": 0,
    "TradeStatus": "1",
  }
}
```

18.付款結果通知

- 應用場景 I：

- Server 端方式(Server POST)([ReturnURL](#))

當消費者付款完成後，特店接受綠界的付款結果訊息，並回應接收訊息

Step1.綠界：以 ServerPost 方式傳送付款結果訊息至特店的 Server 網址([ReturnURL](#))

Step2.廠商：收到綠界的付款結果訊息，回應 1|OK

注意事項：

1.超商繳費條碼(BARCODE)因銀行端作業時間關係會於消費者付款完成兩天後回傳。

2.ATM、CVS、BARCODE 可透過綠界廠商管理後台的『模擬付款』，來確認 ReturnURL 是否正確接收付款結果通知。

3.付款結果通知[常見付款狀態未更新原因說明](#)

4.1|OK 僅是特店回應綠界是否收到通知，並不會改變付款狀態。

- 綠界回傳參數說明 I ([ReturnURL](#))：

- Accept : text/html
 - Content Type : application/json
 - HTTP Method : POST

參數	參數名稱	型態	說明
MerchantID	特店編號	String(10)	特店編號
RpHeader	回傳資料	JSON	
Timestamp	回傳時間	Number	時間戳 Unix timestamp
TransCode	回傳代碼	Int	1 代表 API 傳輸資料(MerchantID, RqHeader, Data)接收成功，實際的 API 執行結果狀態請參考 RtnCode
TransMsg	回傳訊息	String(200)	回傳訊息
Data	加密資料	String	加密過 JSON 格式的資料。 加密方法說明

範例:

```
{
  "MerchantID": "3002607",
  "RpHeader": {
    "Timestamp": 1234567890
  },
  "TransCode": 1,
  "TransMsg": "Success",
  "Data": "..."
```


}

● Data 參數說明：

參數	參數名稱	型態	說明
RtnCode	交易狀態	Int	1 代表 API 執行成功，其餘代碼均為失敗。 失敗代碼請參考 交易訊息代碼表
RtnMsg	回應訊息	String(200)	
RtnCode 為 1 時，以下參數才有值：			
PlatformID	平台商編號	String(10)	特約合作平台商特店編號。
MerchantID	特店編號	String(10)	
SimulatePaid	是否為模擬付款	Int	當交易為模擬付款時，才會回傳此欄位 回傳值： 1:代表此交易為模擬付款，RtnCode 也為 1。並非是由消費者實際真的付款，所以綠界也不會撥款給特店，請勿對該筆交易做出貨等動作，以避免損失。 注意事項： 1.特店可透過綠界廠商後台來針對單筆訂單模擬綠界回傳付款通知，以方便介接 API。 2.此功能僅只是用於測試 ReturnURL 是否能成功接收，不會改變付款狀態。 3.只有透過綠界廠商後台的定期定額查詢功能發動的模擬付款通知，綠界才會傳送此參數，正常由定期定額排程所發送的付款通知，不會傳送此參數。
OrderInfo		JSON Object	
MerchantTradeNo	特店交易編號	String(20)	
TradeNo	綠界的交易編號	String(20)	請保存綠界的交易編號與特店交易編號[MerchantTradeNo]的關聯。
PaymentDate	付款時間	String(20)	格式為 yyyy/MM/dd HH:mm:ss
TradeAmt	交易金額	Int	
PaymentType	特店選擇的付款方式	String(20)	
TradeDate	訂單成立時間	String(20)	yyyy/MM/dd HH:mm:ss
ChargeFee	手續費	Number	
TradeStatus	交易狀態	String(8)	回傳值： 若為 0 時，代表交易訂單成立未付款 若為 1 時，代表交易訂單成立已付款

付款方式為 CVS 時，回傳 CVSInfo				
CVSInfo		CVS 資訊	JSON Object	
	PayFrom	繳費超商	String(10)	family：全家 hilife：萊爾富 okmart：OK 超商 ibon：7-11
	PaymentNo	繳費代碼	String(14)	
	PaymentURL	繳費連結	String(100)	
付款方式為 Barcode 時，回傳 BarcodeInfo				
BarcodeInfo		Barcode 資訊	JSON Object	
	PayFrom	繳費超商	String(10)	family：全家 hilife：萊爾富 okmart：OK 超商 ibon：7-11
付款方式為 ATM 時，回傳 ATMInfo				
ATMInfo		ATM 資訊	JSON Object	
	ATMAccBank	付款人銀行代碼	String(3)	
	ATMAccNo	付款人銀行帳號 後五碼	String(5)	
付款方式為信用卡或銀聯卡時，回傳 CardInfo				
CardInfo		授權資訊	JSON Object	
	AuthCode	銀行授權碼	String(6)	付款方式為銀聯卡時不回傳
	Gwsr	授權交易單號	Int	
	ProcessDate	交易時間	String(20)	yyyy/MM/dd HH:mm:ss
	Amount	金額	Int	
	Eci	3D(VBV) 回傳值	Int	Eci=5,6,2,1 代表該筆交易為 3D 交易 付款方式為銀聯卡時不回傳
	Card6No	信用卡卡號前六碼	String(6)	付款方式為銀聯卡時不回傳
	Card4No	信用卡卡號末四碼	String(4)	付款方式為銀聯卡時不回傳
	Stage	分期期數	Int	分期付款時回傳
	Stast	首期金額	Int	分期付款時回傳
	Staed	各期金額	Int	分期付款時回傳
	RedDan	紅利扣點	Int	紅利折抵時回傳
	RedDeAmt	紅利折抵金額	Int	紅利折抵時回傳

	RedOkAmt	實際扣款金額	Int	紅利折抵時回傳
	RedYet	紅利剩餘點數	Int	紅利折抵時回傳
	PeriodType	訂單建立時的所設定的週期種類	String(1)	定期定額時才會回傳
	Frequency	訂單建立時的所設定的執行頻率	Int	定期定額時才會回傳
	ExecTimes	訂單建立時的所設定的執行次數	Int	定期定額時才會回傳
	PeriodAmount	訂單建立時的每次要授權金額	Int	定期定額時才會回傳
	TotalSuccessTimes	目前已成功授權的次數	Int	定期定額時才會回傳
	TotalSuccessAmount	目前已成功授權的金額合計	Int	定期定額時才會回傳
CoBrandingInfo		聯名卡資料	Array	如果此筆訂單付款的信用卡為特店設定之聯名卡，將會回傳此欄位資料
	CoBrandingCode	聯名卡代號	String(20)	綠界廠商後台設定的聯名卡代號
	Comment	備註	String(100)	
	CustomField	特店自訂欄位	String(200)	提供特店使用客制化欄位

Data 解密後:

```

{
  "RtnCode": 1,
  "RtnMsg": "Success",
  "PlatformID": "1234567890",
  "MerchantID": "1234567890",
  "OrderInfo": {
    "MerchantTradeNo": "test123466",
    "TradeNo": "123456789",
    "TradeDate": "2020/09/18 15:39:10"
  },
  "ThreeDInfo": {
    "ThreeDURL": "https://3durl.com.tw"
  }
}

```

■ Client 端方式([OrderResultURL](#))

當消費者付款完成後，綠界一次性反饋付款結果通知，並將頁面導至特店自製頁面。

Step1.綠界：傳送付款結果並將頁面導至特店的自製頁面網址

Step2.廠商：接收綠界的付款結果訊息。

注意事項：

- 1.若要將付款結果頁顯示於特店自製頁面，請設定[OrderResultURL]
- 2.若有設定此參數，請務必根據回傳的交易狀態來判斷顯示付款成功與否的頁面。
- 3.因各家銀行授權時間不同，若因授權時間過久未收到反饋訊息，請使用查詢訂單 API 查詢後再顯示付款結果。
- 4.若此參數設定網址未使用 https 時，部份瀏覽器可能會出現警告訊息提醒。

● 綠界回傳參數說明 I ([OrderResultURL](#))：

- Accept : text/html
- Content Type : application/x-www-form-urlencoded
- HTTP Method : POST

參數	參數名稱	型態	說明
ResultData	回傳參數	String	廠商自製頁面接收的回傳參數

範例:

ResultData=綠界回傳參數

● ResultData 參數說明：

參數	參數名稱	型態	說明
MerchantID	特店編號	String(10)	特店編號
RpHeader	回傳資料	JSON	
Timestamp	回傳時間	Number	時間戳 Unix timestamp
TransCode	回傳代碼	Int	1 代表傳輸資料(MerchantID, RqHeader, Data)接收成功，其餘均為失敗
TransMsg	回傳訊息	String(200)	回傳訊息
Data	加密資料	String	加密過 JSON 格式的資料。 加密方法說明

範例:

```
{
  "MerchantID": "3002607",
  "RpHeader": {
    "Timestamp": 1234567890,
```

```

},
"TransCode": 1,
"TransMsg": "Success",
"Data": "..."
}

```

● Data 參數說明：

參數	參數名稱	型態	說明
RtnCode	交易狀態	Int	1 代表 API 執行成功，其餘代碼均為失敗。 失敗代碼請參考 交易訊息代碼表
RtnMsg	回應訊息	String(200)	
以下為 RtnCode 為 1 時才會回傳			
PlatformID	平台商編號	String(10)	特約合作平台商特店編號。
MerchantID	特店編號	String(10)	
SimulatePaid	是否為模擬付款	Int	當交易為模擬付款時，才會回傳此欄位 回傳值： 1:代表此交易為模擬付款，RtnCode 也為 1。並非是由消費者實際真的付款，所以綠界也不會撥款給特店，請勿對該筆交易做出貨等動作，以避免損失。 注意事項： 1.特店可透過綠界廠商後台來針對單筆訂單模擬綠界回傳付款通知，以方便介接 API。 2.此功能僅只是用於測試 ReturnURL 是否能成功接收，不會改變付款狀態。 3.只有透過綠界廠商後台的定期定額查詢功能發動的模擬付款通知，綠界才會傳送此參數，正常由定期定額排程所發送的付款通知，不會傳送此參數。
OrderInfo			
	MerchantTradeNo	特店交易編號	String(20)
	TradeNo	綠界的交易編號	String(20) 請保存綠界的交易編號與特店交易編號[MerchantTradeNo]的關聯。
	PaymentDate	付款時間	String(20) 格式為 yyyy/MM/dd HH:mm:ss
	TradeAmt	交易金額	Int
	PaymentType	特店選擇的付款方式	String(20)
	TradeDate	訂單成立時間	String(20) yyyy/MM/dd HH:mm:ss

	ChargeFee	手續費	Number	
	TradeStatus	交易狀態	String(8)	回傳值： 若為 0 時，代表交易訂單成立未付款 若為 1 時，代表交易訂單成立已付款
付款方式為 CVS 時，回傳 CVSInfo				
	CVSInfo	CVS 資訊	JSON Object	
	PayFrom	繳費超商	String(10)	family：全家 hilife：萊爾富 okmart：OK 超商 ibon：7-11
	PaymentNo	繳費代碼	String(14)	
	PaymentURL	繳費連結	String(100)	
付款方式為 Barcode 時，回傳 BarcodeInfo				
	BarcodeInfo	Barcode 資訊	JSON Object	
	PayFrom	繳費超商	String(10)	family：全家 hilife：萊爾富 okmart：OK 超商 ibon：7-11
付款方式為 ATM 時，回傳 ATMInfo				
	ATMInfo	ATM 資訊	JSON Object	
	ATMAccBank	付款人銀行代碼	String(3)	
	ATMAccNo	付款人銀行帳號 後五碼	String(5)	
付款方式為信用卡或銀聯卡時，回傳 CardInfo				
	CardInfo	授權資訊	JSON Object	
	AuthCode	銀行授權碼	String(6)	付款方式為銀聯卡時不回傳
	Gwsr	授權交易單號	Int	
	ProcessDate	交易時間	String(20)	yyyy/MM/dd HH:mm:ss
	Amount	金額	Int	
	Eci	3D(VBV) 回傳值	Int	Eci=5,6,2,1 代表該筆交易為 3D 交易 付款方式為銀聯卡時不回傳
	Card6No	信用卡卡號前六碼	String(6)	付款方式為銀聯卡時不回傳
	Card4No	信用卡卡號末四碼	String(4)	付款方式為銀聯卡時不回傳
	Stage	分期期數	Int	分期付款時回傳

	Stast	首期金額	Int	分期付款時回傳
	Staed	各期金額	Int	分期付款時回傳
	RedDan	紅利扣點	Int	紅利折抵時回傳
	RedDeAmt	紅利折抵金額	Int	紅利折抵時回傳
	RedOkAmt	實際扣款金額	Int	紅利折抵時回傳
	RedYet	紅利剩餘點數	Int	紅利折抵時回傳
	PeriodType	訂單建立時的所設定的週期種類	String(1)	定期定額時才會回傳
	Frequency	訂單建立時的所設定的執行頻率	Int	定期定額時才會回傳
	ExecTimes	訂單建立時的所設定的執行次數	Int	定期定額時才會回傳
	PeriodAmount	訂單建立時的每次要授權金額	Int	定期定額時才會回傳
	TotalSuccessTimes	目前已成功授權的次數	Int	定期定額時才會回傳
	TotalSuccessAmount	目前已成功授權的金額合計	Int	定期定額時才會回傳
CoBrandingInfo		聯名卡資料	Array	
	CoBrandingCode	聯名卡代號	String(20)	
	Comment	備註	String(100)	
CustomField		特店自訂欄位	String(200)	提供特店使用客制化欄位

Data 解密後:

```

{
  "RtnCode": 1,
  "RtnMsg": "Success",
  "PlatformID": "1234567890",
  "MerchantID": "1234567890",
  "OrderInfo": {
    "MerchantTradeNo": "test123466",
    "TradeNo": "123456789",
    "TradeDate": "2020/09/18 15:39:10"
  },
  "ThreeDInfo": {
    "ThreeDURL": "https://3durl.com.tw"
  }
}

```

```

}
}

```

注意事項：

- (1) 若未正確回應 1|OK，系統會隔 5~15 分鐘後重發訊息給特店，當天重複發送四次。
- (2) 若特店持續收到綠界回傳付款資訊，此時請檢查是否未正確回應 1|OK 給綠界，常見錯誤回傳值為(“1|OK”、1|ok、_OK、1\OK、空白)。
- (3) 若遇消費者已付款，但未收到綠界回傳付款完成資訊，此時請檢查接收回傳參數的伺服器是否服務正常，導致無法接收。請參考：[無法收到綠界回傳的付款結果通知](#)
- (4) 特店務必判斷交易狀態[RtnCode]是否為 1，若非 1 時請勿對該筆交易做出貨動作，並取得交易訊息 [RtnMsg] 記錄失敗原因。

● **應用場景 II：**

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，綠界會傳送付款通知結果

Step 1. 綠界：第二次授權是由綠界的排程進行授權，交易授權成功後以 Server POST 方式傳送付款通知至特店傳送的付款完成通知回傳網址[[PeriodReturnURL](#)]。

Step 2. 特店：收到綠界的付款通知訊息後，需比對檢查碼是否相符。

注意事項：

- (1) 第一次授權成功時，付款結果參數會回傳到付款完成通知回傳網址[ReturnURL]。
- (2) 第二次授權成功是由綠界的排程進行授權，所以從第二次開始的資料，都是回傳到所設定的付款完成通知回傳網址[PeriodReturnURL]。

● **綠界回傳參數說明 II：**

- Accept : text/html
- Content Type : application/json
- HTTP Method : POST

參數	參數名稱	型態	說明
MerchantID	特店編號	String(10)	特店編號
RpHeader	回傳資料	JSON	
Timestamp	回傳時間	Number	時間戳 Unix timestamp
TransCode	回傳代碼	Int	1 代表傳輸資料(MerchantID, RqHeader, Data)接收成功，其餘均為失敗
TransMsg	回傳訊息	String(200)	回傳訊息
Data	加密資料	String	加密過 JSON 格式的資料。 加密方法說明

範例:


```
{
  "MerchantID": "3002607",
  "RpHeader": {
    "Timestamp": 1234567890
  },
  "TransCode": 1,
  "TransMsg": "Success",
  "Data": "..."
}
```

● Data 參數說明：

參數	參數名稱	型態	說明
RtnCode	交易狀態	Int	1 代表 API 執行成功，其餘代碼均為失敗。 失敗代碼請參考 交易訊息代碼表
RtnMsg	回應訊息	String(200)	
以下為 RtnCode 為 1 時才會回傳			
PlatformID	平台商編號	String(10)	特約合作平台商特店編號。
MerchantID	特店編號	String(10)	
OrderInfo	訂單資訊	JSON Object	
MerchantTradeNo	特店交易編號	String(20)	
TradeNo	綠界的交易編號	String(20)	請保存綠界的交易編號與特店交易編號[MerchantTradeNo]的關聯。
PaymentDate	付款時間	String(20)	格式為 yyyy/MM/dd HH:mm:ss
TradeAmt	交易金額	Int	
PaymentType	特店選擇的付款方式	String(20)	
TradeDate	訂單成立時間	String(20)	格式為 yyyy/MM/dd HH:mm:ss
ChargeFee	手續費	Number	
TradeStatus	交易狀態	String(8)	回傳值： 若為 0 時，代表交易訂單成立未付款 若為 1 時，代表交易訂單成立已付款
CardInfo	授權資訊	JSON Object	
AuthCode	銀行授權碼	String(6)	
Gwsr	授權交易單號	Int	
ProcessDate	交易時間	String(20)	yyyy/MM/dd HH:mm:ss
Amount	金額	Int	

Eci	3D(VBV) 回傳值	Int	Eci=5,6,2,1 代表該筆交易為 3D 交易
Card6No	信用卡卡號前六碼	String(6)	
Card4No	信用卡卡號末四碼	String(4)	
PeriodType	訂單建立時的所設定的週期種類	String(1)	
Frequency	訂單建立時的所設定的執行頻率	Int	
ExecTimes	訂單建立時的所設定的執行次數	Int	
PeriodAmount	訂單建立時的每次要授權金額	Int	
TotalSuccessTimes	目前已成功授權的次數	Int	
TotalSuccessAmount	目前已成功授權的金額合計	Int	
CustomField	特店自訂欄位	String(200)	提供特店使用客制化欄位

Data 解密後:

```

{
  "RtnCode": 1,
  "RtnMsg": "Success",
  "PlatformID": "1234567890",
  "MerchantID": "1234567890",
  "OrderInfo": {
    "MerchantTradeNo": "test123466",
    "TradeNo": "123456789",
    "TradeDate": "2020/09/18 15:39:10"
  },
  "ThreeDInfo": {
    "ThreeDURL": "https://3durl.com.tw"
  }
}

```

※注意事項：

- (1) [\[PeriodReturnURL\]](#)每期僅會收到一次通知，如未收到通知，請利用[信用卡定期定額訂單查詢](#)取得實際授權結果。
- (2) 若遇消費者已付款，但未收到綠界回傳付款完成資訊，此時請檢查接收回傳參數的伺服器是否服務正常，導致無法接收。請參考：[無法收到綠界回傳的付款結果通知](#)
- (3) 特店務必判斷交易狀態[RtnCode]是否為 1，若非 1 時請勿對該筆交易做出貨動作，並取得交易訊息[RtnMsg]記錄失敗原因。

19.查詢訂單(Server)

- 應用場景:

提供廠商查詢訂單資訊，可透過此 API 來過濾是否為有效訂單。

- 介接路徑：

- 正式環境：https://ecpayment.ecpay.com.tw/1.0.0/Cashier/QueryTrade
- 測試環境：https://ecpayment-stage.ecpay.com.tw/1.0.0/Cashier/QueryTrade

- 特店傳入參數說明：(JSON 格式)

- Content Type : application/json
- HTTP Method : POST

參數	參數名稱	型態	說明
*MerchantID	特店編號	String(10)	特店編號 若為平台商時，參數請帶平台商所綁的特店編號
*RqHeader	傳輸資料	JSON	
	*Timestamp	傳輸時間	Number
			請將傳輸時間轉換為時間戳(GMT+8)，綠界會利用此參數將當下的時間轉為 Unix TimeStamp 來驗證此次介接的時間區間。 注意事項： 若時間戳跟綠界伺服器接收到時間超過 10 分鐘時，交易會失敗無法進行
	*Revision	串接版號	String(10)
			請參考文件封面的 文件版號 。Ex: 1.0.0
*Data	加密資料	String	加密過 JSON 格式的資料。 加密方法說明

範例:

```
{
  "MerchantID": "3002607",
  "RqHeader": {
    "Timestamp": 1234567890,
    "Revision": "1.0.0"
  },
  "Data": "..."
}
```

- Data 參數說明：請在加密前對參數值進行 URLEncode

參數	參數名稱	型態	說明
PlatformID	平台商代號	String(10)	平台商特店編號
*MerchantID	特店編號	String(10)	
*MerchantTradeNo	特店交易編號	String(20)	

Data 範例：

```
{
  "MerchantID": "3002607",
  "MerchantTradeNo": "20180914001"
}
```

- 綠界回傳參數說明：(JSON 格式)

參數	參數名稱	型態	說明
MerchantID	特店編號	String(10)	
RpHeader	回傳資料	JSON	
Timestamp	回傳時間	Number	時間戳 Unix timestamp
TransCode	回傳代碼	Int	1 代表 API 傳輸資料(MerchantID, RqHeader, Data)接收成功，實際的 API 執行結果狀態請參考 RtnCode
TransMsg	回傳訊息	String(200)	回傳訊息
Data	加密資料	String	加密過 JSON 格式的資料。 加密方法說明

範例:

```
{
  "MerchantID": "3002607",
  "RpHeader": {
    "Timestamp": 1234567890
  },
  "TransCode": 1,
  "TransMsg": "成功",
  "Data": "..."
}
```

● Data 參數說明：請在解密後對參數值進行 URLDecode

參數	參數名稱	型態	說明
RtnCode	交易狀態	Int	1 代表 API 執行成功，其餘代碼均為失敗。 失敗代碼請參考 交易訊息代碼表
RtnMsg	回應訊息	String(200)	
RtnCode 為 1 時，以下參數才有值：			
PlatformID	平台商代號	String(10)	平台商特店編號
MerchantID	特店編號	String(10)	
OrderInfo	訂單資訊	JSON	
MerchantTradeNo	特店交易編號	String(20)	不可重複
TradeNo	綠界交易編號	String(20)	
TradeAmt	交易金額	Int	
TradeDate	訂單成立時間	String(20)	格式為 yyyy/MM/dd HH:mm:ss
PaymentType	付款方式	String(20)	
PaymentDate	付款時間	String(20)	格式為 yyyy/MM/dd HH:mm:ss
ChargeFee	手續費	Number	
TradeStatus	交易狀態	String(8)	回傳值： 若為 0 時，代表交易訂單付款失敗 若為 1 時，代表交易訂單付款成功
CardInfo	授權資訊	JSON	
AuthCode	銀行授權碼	String(6)	
Gwsr	授權交易單號	Int	
ProcessDate	交易時間	String(20)	yyyy/MM/dd HH:mm:ss
Amount	金額	Int	
Stage	分期期數	Int	
Stast	首期金額	Int	
Staed	各期金額	Int	
Eci	3D(VBV) 回傳值	Int	Eci=5,6,2,1 代表該筆交易為 3D 交易
Card6No	信用卡卡號前六碼	String(6)	
Card4No	信用卡卡號末四碼	String(4)	
RedDan	紅利扣點	Int	紅利折抵時回傳
RedOkAmt	實際扣款金額	Int	紅利折抵時回傳
RedYet	紅利剩餘點數	Int	紅利折抵時回傳
PeriodType	週期種類	String (1)	定期定額時回傳，訂單建立時所設定的週期種類

				D：以天為週期・M：以月為週期・Y：以年為週期
	Frequency	執行頻率	Int	定期定額時回傳・訂單建立時所設定的執行頻率
	ExecTimes	執行次數	Int	定期定額時回傳・訂單建立時所設定的執行次數
	PeriodAmount	訂單建立時的每次要授權金額	Int	定期定額時回傳
	TotalSuccessTimes	目前已成功授權的次數	Int	定期定額時回傳・目前已成功授權的次數。
	TotalSuccessAmount	目前已成功授權的金額合計	Int	定期定額時回傳
CVSInfo		CVS 資訊	JSON	
	PayFrom	繳費超商	String(10)	
	PaymentNo	繳費代碼	String(14)	
	PaymentURL	繳費連結	String(14)	
BarcodeInfo		Barcode 資訊	JSON	
	PayFrom	繳費超商	String(10)	
ATMInfo		ATM 資訊	JSON	
	ATMAccBank	付款人銀行代碼	String(3)	
	ATMAccNo	付款人銀行帳號後五碼	String(5)	
CoBrandingInfo		聯名卡資料	Array	如果此筆訂單付款的信用卡為特店設定之聯名卡・將會回傳此欄位資料
	CoBrandingCode	聯名卡代號	String(20)	綠界廠商後台設定的聯名卡代號
	Comment	備註	String(100)	
CustomField		特店自訂欄位	String(200)	提供特店使用客制化欄位

Data 範例：

```

{
  "RtnCode": 1,
  "RtnMsg": "Success",
  "MerchantID": "3002607",
  "OrderInfo": {
    "MerchantTradeNo": "20180914001",
    "TradeNo": "1809261503338172",
    "TradeDate": "2018/09/26 14:59:54"
  },
  "CardInfo": {

```

```
"Gwsr": 10735183,  
"ProcessDate": "2018/09/26 14:59:54",  
"AuthCode": "777777",  
"Amount": 100,  
"Eci": 2,  
"Card4No": "2222",  
"Card6No": "491122",  
"RedDan": 0,  
"RedOkAmt": 0,  
"RedYet": 0  
}  
}
```


20.查詢信用卡單筆明細記錄(Server)

- 應用場景：

提供查詢信用卡交易的單筆明細記錄。

- 介接路徑：

- 正式環境：<https://ecpayment.ecpay.com.tw/1.0.0/CreditDetail/QueryTrade>
- 測試環境：<https://ecpayment-stage.ecpay.com.tw/1.0.0/CreditDetail/QueryTrade>

- 特店傳入參數說明：(JSON 格式)

- Content Type : application/json
- HTTP Method : POST

參數	參數名稱	型態	說明
*MerchantID	特店編號	String(10)	特店編號 若為平台商時，參數請帶平台商所綁的特店編號
*RqHeader	傳輸資料	JSON	
*Timestamp	傳輸時間	Number	請將傳輸時間轉換為時間戳(GMT+8)，綠界會利用此參數將當下的時間轉為 Unix TimeStamp 來驗證此次介接的時間區間。 注意事項： 若時間戳跟綠界伺服器接收到時間超過 10 分鐘時，交易會失敗無法進行
*Revision	串接版號	String(10)	請參考文件封面的 文件版號 。Ex: 1.0.0
*Data	加密資料	String	加密過 JSON 格式的資料。 加密方法說明

範例:

```
{  
  "MerchantID": "3002607",  
  "RqHeader": {  
    "Timestamp": 1234567890,  
    "Revision": "1.0.0"  
  },  
  "Data": "..."  
}
```

- Data 參數說明：請在加密前對參數值進行 URLEncode

參數	參數名稱	型態	說明
PlatformID	平台商編號	String(10)	特約合作平台商特店編號
*MerchantID	特店編號	String(10)	
*MerchantTradeNo	特店交易編號	String(20)	
TradeNo	綠界交易編號	String(20)	此欄位可從 定期定額訂單查詢 API 取得。可查詢定期定額第一筆之後的授權資料內容。

Data 範例：

```
{
  "MerchantID": "3002607",
  "PlatformID": "2000132",
  "MerchantTradeNo": "123486456"
}
```

- 綠界回傳參數說明：(JSON 格式)

參數	參數名稱	型態	說明
MerchantID	特店編號	String(10)	特店編號
RpHeader	傳輸資料	JSON	
Timestamp	傳輸時間	Number	時間戳 Unix timestamp 注意事項： 若時間戳跟綠界伺服器接收到時間超過 10 分鐘時，交易會失敗無法進行
TransCode	回傳代碼	Int	1 代表 API 傳輸資料(MerchantID, RqHeader, Data)接收成功，實際的 API 執行結果狀態請參考 RtnMsg
TransMsg	回傳訊息	String(200)	回傳訊息
Data	加密資料	String	加密過 JSON 格式的資料。 加密方法說明

範例：

```
{
  "MerchantID": "3002607",
  "RpHeader": {
    "Timestamp": 1235467890
  },
  "TransCode": 1,
  "TransMsg": "Success",
  "Data": "..."
}
```

```
"Data": "..."  
}
```

- Data 參數說明：請在解密後對參數值進行 URLDecode

參數	參數名稱	型態	說明
RtnMsg	回應訊息	String	若成功回傳空值其餘則為失敗 若查無商家或商家已到期，回傳值為：error_Stop 若查無該筆交易授權單號時，回傳值為：error_nopay 若發生錯誤或資料檢核失敗，則回傳值為：error 若發生傳遞的資料長度不對，則回傳值為：error_Length 若發生金額參數不對，則回傳值為：error_Amount 若發生該筆交易帳目運作中，無法由程式關帳，則回傳值為：error_close
RtnValue	回應資料	JSON	RtnMsg 為空值時才會回傳
RtnValue 回應內容			
TradeID	授權單號	Int	
Amount	交易金額	Int	
ClsAmt	已關帳金額	Int	
AuthTime	訂單成立時間	String(24)	
Status	交易狀態	String(30)	若未有關帳明細交易，狀態回傳值如下： 已取消(Canceled)：此筆交易已取消 未授權(Unauthorized)：銀行未授權完成 已授權(Authorized)：銀行已完成授權 若已有關帳明細交易，狀態回傳值如下： 銀行拒絕(Bank Refused) 要關帳(To be captured) 關帳中(Captured in progress) 已關帳(Captured) 要取消(To be canceled) 取消中(Canceled in progress) 銀行追回中(Bank chargebacked in progress) 銀行已追回(Bank chargebacked) 不明(Unknown) 操作取消(Operation canceled)
CloseData	交易明細	Array	
	Status	狀態	狀態回傳值如下： 銀行拒絕(Bank Refused) 取消中(Canceled in progress)

				銀行追回中(Bank chargebacked in progress) 銀行已追回(Bank chargebacked) 批次失敗(Batch failed) 不明(Unknown) 操作取消(Operation canceled)
	Amount	交易金額	Int	
	DateTime	關帳時間	String(24)	

Data 解密後：

```
{
  "RtnMsg": "",
  "RtnValue": {
    "TradeID ": 245678,
    "Amount ":100,
    "ClsAmt ": 100,
    "AuthTime": "2018/09/25 14:59:54",
    "Status": "To be captured"
  },
  "CloseData": [{
    "Status": "Bank chargebacked ",
    "Amount": 100,
    "DateTime" : "2018/09/26 14:59:54"
  }]
}
```

21.

21.信用卡請退款功能(Server)

- 應用場景：

當收到訂單授權成功，透由[查詢信用卡單筆明細記錄 API](#) 取得[交易狀態]，再依[交易狀態]發動欲執行動作。

1. 請款使用說明:信用卡付款成功，查詢訂單會顯示[已授權] (此時僅會佔用信用卡額度)，如果此筆訂單要送至銀行請款，須執行[關帳](Action=C)。
2. 執行[關帳]後，查詢訂單會顯示[要關帳]，表示綠界將要向銀行進行請款作業。
3. 當綠界已向銀行完成[請款]作業，查詢訂單會顯示[已關帳]。

- 退款使用方式:

1. 呼叫[查詢信用卡單筆明細紀錄 API](#) 取得狀態
2. 查詢後，呼叫信用卡請退款 API:
 - [已授權]階段: 執行[放棄] (Action=N)可釋放信用卡佔額。
 - [要關帳]階段:
 - i. 全額退款: 先執行[取消] (Action=E)，接著進行[放棄] (Action=N)。
 - ii. 部份退款: 執行[退刷] (Action=R)。
 - [已關帳]階段: 執行[退刷] (Action=R)。

- 執行動作說明:

■ 「關帳」：訂單在每日 20:00 系統會自行關帳，綠界依關帳後的金額向銀行請/退款。

注意事項：

- (1) 若於綠界廠商後台將系統自動關帳關閉以後，訂單仍須於 21 天內完成關帳作業，若無後續將無法以 API 方式進行關帳，錯誤訊息-error_overDAY 須洽客服手動關帳；若未對訂單進行關帳，第 80 天時將發送信用卡未完成關帳通知；超過 90 天系統會將未關帳訂單進行放棄作業。
- (2) 如付款方式為銀聯卡，關帳方式為立即關帳。

- 「退刷」：在系統關帳或自行關帳後，可修改訂單金額。

※修改上限為訂單金額

注意事項：

- (1) 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。
- (2) 使用紅利折抵交易得全額退刷，僅一般交易才可退刷部分金額。
- (3) 如帳戶餘額低於退刷金額，將無法退刷，建議留存一定金額於綠界帳戶或至綠界廠商後台預存綠界帳戶餘額供退刷之用。

- 「取消」：可取消關帳，讓訂單回復關帳前的狀態。
- 「放棄」：適用在當日關帳前放棄此筆交易，不做請款。

注意事項：

- (1) 此 API 不支援定期定額「停用收款」功能，若要停用定期定額收款請登入綠界廠商後台進行操作，系統路徑為信用卡收單>定期定額查詢>明細 / 編輯。
- (2) 不支援部分金額退款，一律會以訂單全額進行退款。

● 介接路徑：

- 正式環境：<https://ecpayment.ecpay.com.tw/1.0.0/Credit/DoAction>
- 測試環境：因無法提供實際授權，故無法使用此 API

● 廠商傳入參數說明：(JSON 格式)

- Content Type : application/json
- HTTP Method : POST

參數	參數名稱	型態	說明
*MerchantID	特店編號	String(10)	特店編號 若為平台商時，參數請帶平台商所綁的特店編號
*RqHeader	傳輸資料	JSON	
*Timestamp	傳輸時間	Number	請將傳輸時間轉換為時間戳(GMT+8)，綠界會利用此參數將當下的時間轉為 Unix TimeStamp 來驗證此次介接的時間區間。 注意事項： 若時間戳跟綠界伺服器接收到時間超過 10 分鐘時，交易會失敗無法進行。
*Revision	串接版號	String(10)	請參考文件封面的 文件版號 。Ex: 1.0.0
*Data	加密資料	String	加密過 JSON 格式的資料。 加密方法說明

範例:

```
{
  "MerchantID": "3002607",
  "RqHeader": {
    "Timestamp": 1234567890,
    "Revision": "2.0.0"
  },
  "Data": "..."
}
```

- Data 參數說明：請在加密前對參數值進行 URLEncode

參數	參數名稱	型態	說明
PlatformID	平台商代號	String(10)	平台商特店編號
*MerchantID	特店編號	String(10)	綠界提供
*MerchantTradeNo	特店交易編號	String(20)	不可重覆
*TradeNo	綠界交易編號	String(20)	
*Action	執行方式	String(1)	針對訂單做處理的動作，例如要對訂單做關帳時，請帶 C。 C:關帳 R:退刷 E:取消 N:放棄
*TotalAmount	金額	Int	

Data 範例：

```
{
  "MerchantID": "3002607",
  "MerchantTradeNo": "20180914001",
  "TradeNo": "1809261503338172",
  "Action": "C",
  "TotalAmount": 100
}
```

- 綠界回傳參數說明：(JSON 格式)

參數	參數名稱	型態	說明
MerchantID	特店編號	String(10)	
RpHeader	回傳資料	JSON	
Timestamp	回傳時間	Number	時間戳 Unix timestamp
TransCode	回傳代碼	Int	1 代表 API 傳輸資料(MerchantID, RpHeader, Data)接收成功，實際的 API 執行結果狀態請參考 RtnCode
TransMsg	回傳訊息	String(200)	回傳訊息
Data	加密資料	String	加密過 JSON 格式的資料。 加密方法說明

範例:

```
{
  "MerchantID": "3002607",
  "RpHeader": {
    "Timestamp": 1234567890
  },
  "TransCode": 1,
  "TransMsg": "成功",
  "Data": "..."
}
```

- Data 參數說明：請在解密後對參數值進行 URLDecode

參數	參數名稱	型態	說明
RtnCode	執行方式	Int	1 代表 API 執行成功，其餘代碼均為失敗。 失敗代碼請參考 交易訊息代碼表
RtnMsg	交易訊息	String(200)	
以下為 RtnCode 為 1 時才會回傳			
PlatformID	平台商代號	String(10)	平台商特店編號
MerchantID	特店編號	String(10)	
MerchantTradeNo	特店交易編號	String(20)	不可重覆
TradeNo	綠界交易編號	String(20)	

Data 範例：

```
{  
  "MerchantID": "3002607",  
  "MerchantTradeNo": "20180914001",  
  "TradeNo": "1809261503338172",  
  "RtnCode": 1,  
  "RtnMsg": ""  
}
```

附錄 1. 參數加密方式說明

依提供 AES 加解密用的 Key 及 IV，請將要加密的資料做 URL Encode 編碼，再進行 AES 加密
AES 加密的強度設定方式是 128 bit，CipherMode : CBC，PaddingMode:PKCS7

- 加密範例：Key=A123456789012345，IV=B123456789012345

(1)加密前 Data 資料：

```
{"Name":"Test","ID":"A123456789"}
```

(2)URLEncode 編碼後結果：

```
%7B%22Name%22%3A%22Test%22%2C%22ID%22%3A%22A123456789%22%7D
```

(3)AES 加密後結果：

```
7woM9RorZKAtXJRVccAb0qhHYm+5lnlhBzyfh5EZdNck7PacNsRHgv/Jvp//ajJidqcQcs0UmAgPQVjXQHeziw==
```

- 解密範例：

(1)Data 密文：

```
7woM9RorZKAtXJRVccAb0qhHYm+5lnlhBzyfh5EZdNck7PacNsRHgv/Jvp//ajJidqcQcs0UmAgPQVjXQHeziw==
```

(2)AES 解密結果：

```
%7B%22Name%22%3A%22Test%22%2C%22ID%22%3A%22A123456789%22%7D
```

(3)URLDecode 解碼後結果：

```
{"Name":"Test","ID":"A123456789"}
```

附錄 2. 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，請到綠界廠商管理後台->系統開發管理->交易狀態代碼查詢。

綠界廠商管理後台

合約商議 回首頁 客服信箱

綠界客服專線：(02) 2655-1775
服務的時間：週一至週日 09:00-18:00

交易狀態代碼查詢

系統開發管理

交易狀態代碼查詢

您可於此功能下查看程式回傳的交易狀態代碼說明。

交易狀態代碼

請輸入交易狀態代碼

每頁顯示 10筆

查詢

附錄 3. CSS 參數定義列表

Class	說明
ecpay-MAIN-CONTENT	信用卡付款畫面
ecpay-pay-list-wrap	付款選擇頁
ecpay-col-a-100	
ecpay-col-d-50	
ecpay-red	
ecpay-card-info	
ecpay-pay-check	記憶卡號功能畫面
ecpay-card-info-tab	信用卡卡號輸入畫面
ecpay-cit-item-box	
ecpay-select-input	
ecpay-add-box	新增信用卡卡號輸入框
ecpay-edit-box	編輯信用卡卡號輸入框
ecpay-result-block	新增/修改結果畫面
ecpay-ad-wrap	
ecpay-order-title	
ecpay-order-info	
ecpay-order-payway	
ecpay-oi-pay-content	
ecpay-btn-box	
ecpay-btn	

附錄 4. 銀行代碼表

付款方式	銀行名稱	銀行代碼
ATM	ATM_台新銀行	812
	ATM_玉山(暫不提供)	808
	ATM_台灣銀行	004
	ATM_台北富邦(暫不提供)	813
	ATM_中國信託	822
	ATM_第一銀行	007
	ATM_土地銀行	005
	ATM_國泰世華銀行(暫不提供)	827
	ATM_大眾銀行	814
	ATM_板信銀行	118