



綠界科技股份有限公司

Green World FinTech Service Co., Ltd.

信用卡 APP SDK

介接技術文件

V 1.0.0

文件編號 gw_p603

2020-08-19

Version History :

Version	Date	Content
V1.0.0	2020/08/19	Create

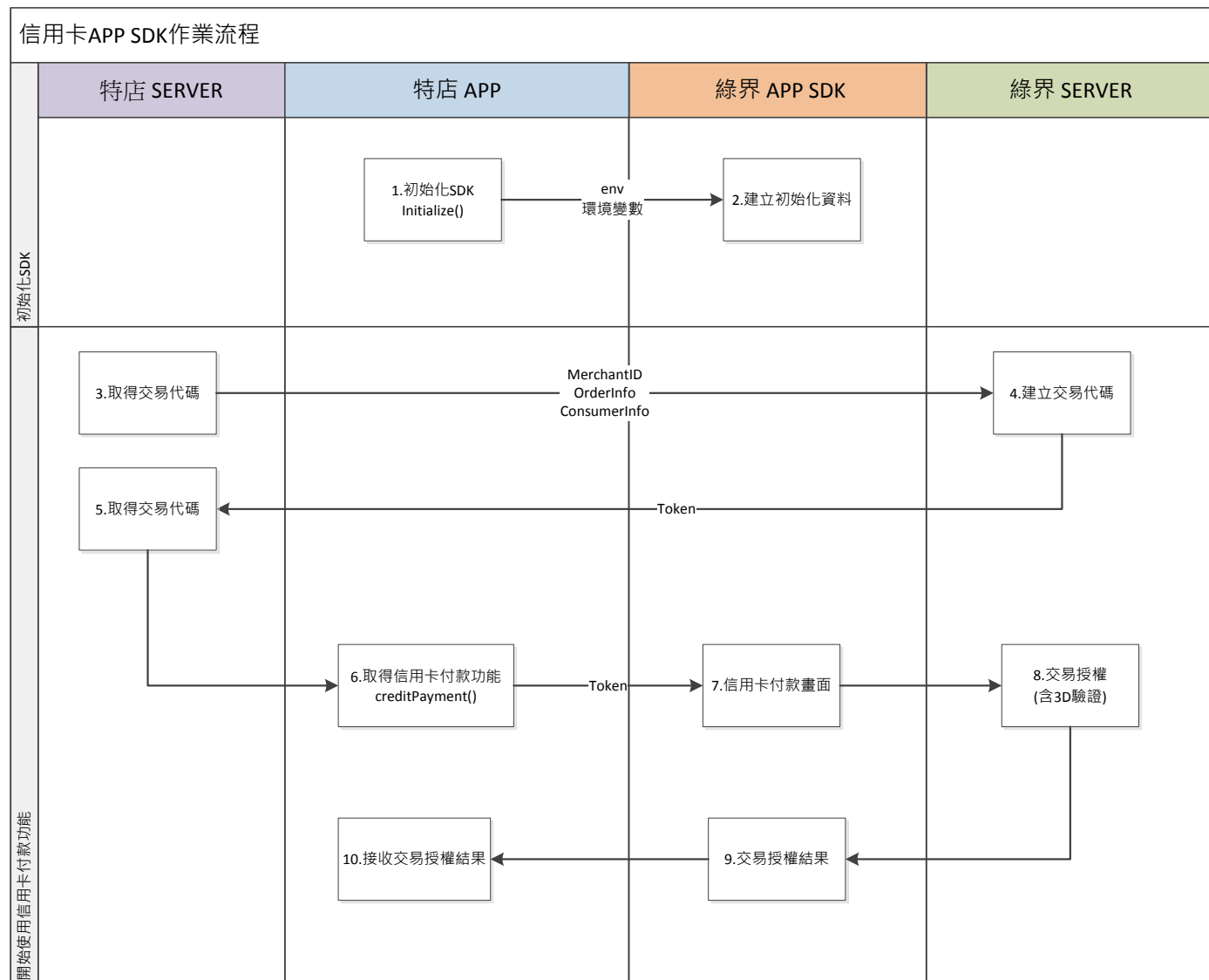
目錄

1. 簡介.....	4
2. 信用卡 APP SDK 流程使用說明	5
3. 系統介接測試相關資訊.....	7
4. 取得交易代碼(Server)	8
5. 信用卡付款 Android SDK(APP)	12
6. 信用卡付款 iOS SDK(APP)	15
7. 查詢訂單(Server)	18
8. 查詢信用卡單筆明細記錄(Server)	22
9. 信用卡請退款功能(Server)	27
附錄 1. 參數加密方式說明	31
附錄 2. 交易訊息代碼一覽表	32

1. 簡介

提供特店使用 APP SDK 串接綠界信用卡服務。特店可利用 SDK 產生的信用卡功能畫面進行交易。此規格也提供特店透過 Server 查詢交易明細與信用卡請退款功能。

2. 信用卡 APP SDK 流程使用說明



處理角色	流程名稱	處理說明
特店 APP	1.初始化 SDK	開始使用 SDK 需先初始化，並帶入環境變數
綠界 APP SDK	2.建立 SDK 初始化資料	SDK 建立初始化資料
特店 Server	3.取得 交易代碼	特店確定訂單內容後，傳送訂單資料到綠界並取得交易代碼
綠界 Server	4.建立交易代碼	綠界 Server 接收到訂單資料後，產生一組交易代碼
特店 Server	5.取得交易代碼	接收綠界回傳的交易代碼
特店 APP	6.取得信用卡付款功能	呼叫取得信用卡付款功能
綠界 APP SDK	7.信用卡付款畫面	產生信用卡付款畫面
特店 Server	8.交易授權(含 3D 驗證)	將信用卡資料送至綠界 Server 建立訂單並將信用卡資料送至銀行進行授權與 3D 驗證
綠界 APP SDK	9.交易授權結果	綠界 APP SDK 接收交易授權結果，並將結果回傳至特店 APP
特店 APP	10.接收交易授權結果	接收 SDK 回傳的交易授權結果

3. 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，測試無誤要連正式環境時，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	--
平台商編號(PlatformID)	--	3003008
登入廠商後台帳號/密碼	stagetest3/test1234	StageTestV3P/test1234
統一編號/身分證號	00000000	10608171
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	FCnGLNS7P3xQ2q3E
all in one 介接的 HashIV	EkRm7iFT261dpevs	awL5GRWRhyaybq13
信用卡測試卡號	4311-9522-2222-2222	
信用卡測試安全碼	222	
信用卡測試有效年月	請設定大於測試時間。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： <ol style="list-style-type: none"> 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： 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。 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。 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。 請確認 Server URL 連接 port 為 http 80 port 與 https 443 port。 測試環境信用卡付款方式未提供海外卡及銀聯卡服務。 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。 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。 為保障消費者權益與網路交易安全，串接服務只支援 TLS 1.1 以上之加密通訊協定。 		

4. 取得交易代碼(Server)

● 應用場景：

特店在使用綠界信用卡 APP SDK 前，特店 Server 必須先向綠界 Server 取得交易代碼。

● 介接路徑

正式環境：<https://proxy.ecpay.com.tw/api/token>

測試環境：<https://proxy-stage.ecpay.com.tw/api/token>

● 特店傳入參數說明：(JSON 格式)

參數	參數名稱	型態	說明
*MerchantID	特店編號	String(10)	特店編號
*RqHeader	傳輸資料	JSON	
	*Timestamp	Int	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
	*Revision	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
*MerchantID	特店編號	String(10)	
*AppStoreName	APP 商店名稱	String(20)	顯示於信用卡刷卡畫面的 APP 商店名稱,特店可自行定義顯示的名稱
*OrderInfo	交易資訊	JSON	
*MerchantTradeNo	特店交易編號	String(20)	不可重覆
*MerchantTradeDate	特店交易時間	String(20)	格式為 yyyy/MM/dd HH:mm:ss
*TotalAmount	交易金額	Int	
TradeDesc	交易描述	String(200)	
*ConsumerInfo	消費者資訊	JSON	
*Email	電子信箱	String(50)	持卡人電子信箱
*Phone	電話	String(20)	持卡人手機號碼
*Name	姓名	String(50)	持卡人姓名
*CountryCode	國別碼	String(3)	持卡人帳單地址國別碼·請參考 ISO-3166 編碼 傳送 3 位數字代碼格式
Address	地址	String(50)	持卡人帳單地址

Data 範例：

```
{
  "MerchantID": "3002607",
  "AppStoreName": "綠界商城"
  "OrderInfo": {
    "MerchantTradeNo": "20180914001",
    "MerchantTradeDate": "2020/09/26 14:49:12",
    "TotalAmount": 100
  },
  "ConsumerInfo": {
    "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	Int	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": 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，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			
Token	交易代碼	String(64)	30 分鐘時效
OrderInfo	交易資訊	JSON	
	MerchantTradeNo	String(20)	不可重覆
	MerchantTradeDate	String(20)	格式為 yyyy/MM/dd HH:mm:ss
	TotalAmount	Int	
	TradeDesc	String(200)	

Data 解密後:

```
{
  "RtnCode": 1,
  "RtnMsg": "Success",
  "Token": "12324232ABDGeQJ1232Ddwedew213213213dwdc",
  "OrderInfo": {
    "MerchantTradeNo": "20180914001",
    "MerchantTradeDate": "2020/09/26 14:49:12",
    "TotalAmount": 100,
    "TradeDesc": "..."
  }
}
```

5. 信用卡付款 Android SDK(APP)

● 應用場景:

適用於前端 Android APP，提供特店信用卡刷卡功能。Android SDK 詳細的使用方式與範例程式請參閱 [GitHub](#)

● 安裝方式:

1. APP 專案中 app/build.gradle 檔案新增 dependency

```
dependencies {  
    // ECPay Payment SDK  
    implementation 'tw.com.ecpay.paymentkit:ECPayPaymentKit:1.0.0'  
}
```

2. APP 專案中 project/build.gradle 檔案加入 [JCenter](#)

```
buildscript {  
    repositories {  
        jcenter()  
    }  
}
```

● 使用方式:

1. **Import:**

```
import tw.com.ecpay.paymentkit.manager.*
```

2. **Initialize:** 初始化 SDK，傳入 Activity 與環境參數

```
PaymentkitManager.initialize(Activity mActivity, String env);
```

➤ Input:

參數	參數名稱	型態	說明
*Activity	Activity 元件	Android Activity	請提供 APP 當前畫面的 Activity 元件
*Env	環境參數	String	Stage:測試環境 Prod:正式環境

4. CreditPayment: 產生信用卡刷卡功能畫面

```
PaymentkitManager.creditPayment(Activity, Fragment, Token, MerchantTradeNo, RequestCode);
```

➤ Input:

參數	參數名稱	型態	說明
*Activity	Activity 元件	Android Activity	請提供 APP 當前畫面的 Activity 元件
Fragment	Fragment 元件	Android Fragment	如果 APP 當前畫面的切換方式為 Fragment Fragment 為必填
*Token	交易代碼	String(64)	交易代碼
*MerchantTradeNo	特店交易編號	String(20)	
*RequestCode	畫面返回值	Int	當信用卡刷卡畫面關閉時,返回當前 Activity/Fragment 的值

➤ onActivityResult:

```
PaymentkitManager.creditPaymentResult(mActivity, resultCode, data, new
CallbackFunction<CreditPaymentCallbackData>() {
    @Override public void callback(CreditPaymentCallbackData callbackData) {}
});
```

➤ Callback:

參數	參數名稱	型態	說明
CallbackStatus	Callback 狀態	CallbackStatus	Success:執行成功 Fail:執行失敗 Cancel:取消 Unknown:未知的錯誤
callbackStateMessage	Callback 訊息	String	
以下為 callbackStatus 為 Success 時才會回傳			
RtnCode	回應代碼	Int	1 表示成功 · 其餘代碼表示失敗
RtnMsg	回應訊息	String(200)	
以下為 RtnCode 為 1 時才會回傳			
MerchantID	特店編號	String(10)	
OrderInfo	訂單資訊	JSON	
MerchantTradeNo	廠商交易編號	String(20)	
TradeNo	綠界交易編號	String(20)	
TradeDate	訂單成立時間	String(20)	
CardInfo	信用卡資訊	JSON	
AuthCode	信用卡授權碼	String(6)	
Gwsr	授權交易單號	Int	
ProcessDate	交易時間	String(20)	格式為 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)	
以下為 RtnCode 不為 1 時回傳			
MerchantID	特店編號	String(10)	
OrderInfo	訂單資訊	JSON	
MerchantTradeNo	廠商交易編號	String(20)	

6. 信用卡付款 iOS SDK(APP)

- 應用場景:

適用於前端 iOS APP，提供特店信用卡刷卡功能。iOS SDK 詳細的使用方式與範例程式請參閱 [GitHub](#)

- 支援:

1. Xcode 11.2.1+
2. Swift 4.2+
3. iOS 10.0+

- 安裝方式:

1. [CocoaPods](#):

Podfile 內容:

```
pod 'ECPayPaymentKit', '~> 0.0.1'
```

此套件相依其他 CocoaPods 套件，詳細清單如下

```
pod 'PromiseKit', '~> 6.8.3'
```

2. [Carthage](#):

Cartfile 內容:

```
binary "https://github.com/ECPay/ECPayPaymentKit_iOS/releases/download/0.0.1/ECPayPaymentKit.json" ~> 0.0.1
```

此套件相依其他 Carthage 套件，詳細清單如下:

```
github "mxcl/PromiseKit" ~> 6.8.3
```

- 使用方式:

1. **Import:**

```
import ECPayPaymentKit
```

2. **Initialize:** 建議於 AppDelegate.swift 內執行初始化的動作，環境參數請參考下段說明。

```
ECPayPaymentManager.sharedInstance().initialize(env: .Stage)
```

➤ Input:

參數	參數名稱	型態	說明
* Env	環境參數	String	Stage:測試環境 Prod:正式環境

3. **CreditPayment:** 產生信用卡刷卡功能畫面

```
ECPayPaymentManager.sharedInstance().creditPayment(token: token, merchantTradeNo: merchantTradeNo)
{ (state) in switch state.callbackStateStatus {
    case .Success:
        print("Success")
    case .Fail:
        print("Fail")
    case .Cancel:
        print("Cancel")
    case .Unknown:
        print("Unknown")
}
}
```

➤ Input:

參數	參數名稱	型態	說明
*Token	交易代碼	String(64)	交易代碼
*MerchantTradeNo	特店交易編號	String(20)	
*Callback	Callback	Callback State	此非同步工作執行完成後的後續邏輯處理

➤ Callback:

參數	參數名稱	型態	說明
callbackStatus	Callback 狀態	CallbackStateStatusType	Success:執行成功 Fail:執行失敗 Cancel:取消 Unknown:未知的錯誤
callbackStateMessage	Callback 訊息	String	
以下為 callbackStatus 為 Success 時才會回傳			
RtnCode	回應代碼	Int	1 表示成功，其餘代碼表示失敗
RtnMsg	回應訊息	String(200)	
以下為 RtnCode 為 1 時才會回傳			
MerchantID	特店編號	String(10)	
OrderInfo	訂單資訊	JSON	
MerchantTradeNo	廠商交易編號	String(20)	
TradeNo	綠界交易編號	String(20)	
TradeDate	訂單成立時間	String(20)	
CardInfo	信用卡資訊	JSON	
AuthCode	信用卡授權碼	String(6)	
Gwsr	授權交易單號	Int	
ProcessDate	交易時間	String(20)	格式為 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)	
以下為 RtnCode 不為 1 時回傳			
MerchantID	特店編號	String(10)	
OrderInfo	訂單資訊	JSON	
MerchantTradeNo	廠商交易編號	String(20)	

7. 查詢訂單(Server)

● 應用場景:

提供特店查詢訂單資訊，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)

參數	參數名稱	型態	說明
*MerchantID	特店編號	String(10)	綠界提供
*RqHeader	傳輸資料	JSON	
	*Timestamp	Int	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
	*Revision	String(10)	請參考文件封面的 文件版號 。Ex: 1.0.0
*Data	加密資料		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	Int	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		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": "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)	
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)	
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)	yyyy/MM/dd HH:mm:ss
PaymentType	付款方式	String(20)	Credit
PaymentDate	付款時間	String(20)	yyyy/MM/dd HH:mm:ss
ChargeFee	手續費	Number	格式為整數 8 位,小數 2 位
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
CustomField	特店自訂欄位	String(200)		提供特店使用客制化欄位 JSON

Data 範例：

```

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

```

8. 查詢信用卡單筆明細記錄(Server)

- 應用場景：

提供查詢信用卡明細記錄。

- 介接路徑：

- 正式環境：<http://ecpayment.ecpay.com.tw/1.0.0/CreditDetail/QueryTrade>

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

- 特店 POST 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
*RqHeader	傳輸資料	JSON	
*Timestamp	傳輸時間	Int	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
*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	Int	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": 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	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 若發生該筆交易已帳目運作中，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	String(30) 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"
  }]
}
    
```

9. 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。

- 請款使用說明:

1. 信用卡付款成功，查詢訂單會顯示[已授權] (此時僅會占用信用卡額度)，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：若於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。

- 「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：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。

● 介接路徑：

正式環境：<https://ecpayment.ecpay.com.tw/1.0.0/Credit/DoAction>

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

● 特店傳入參數說明：(JSON 格式)

參數	參數名稱	型態	說明
*MerchantID	特店編號	String(10)	特店編號
*RqHeader	傳輸資料	JSON	
*Timestamp	傳輸時間	Int	時間戳 Unix timestamp 注意事項： 請留存此欄位編號,之後如需與綠界客服人員追查 API 相關問題,請以此編號作為查詢依據。
*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		加密過 JSON 格式的資料。 加密方法說明

範例:

```
{
  "MerchantID": "3002607",
  "RqHeader": {
    "Timestamp": 1525168923,
    "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	Int	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		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": "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，請到廠商後台->系統開發管理->交易狀態代碼查詢。

The screenshot displays the '綠界廠商管理後台' (ECPay Merchant Management Backend) interface. The top navigation bar includes '合約商議', '回首頁', and '客服信箱' icons, along with the contact information: '綠界客服專線：(02) 2655-1775' and '服務的時間：週一至週日 09:00-18:00'. The left sidebar lists various management functions, with '交易狀態代碼查詢' (Transaction Status Code Query) highlighted in a red box. The main content area is titled '系統開發管理' (System Development Management) and contains the '交易狀態代碼查詢' (Transaction Status Code Query) section. This section includes a text input field labeled '交易狀態代碼' with the placeholder '請輸入交易狀態代碼', a dropdown menu for '每頁顯示' (Items per page) set to '10筆', and a green '查詢' (Query) button.