



Green World FinTech Service Co., Ltd.

# **Apple Pay API Document**

V 1.0.0  
2021-03-04

## Version History

Version	Date	Content
V1.0.0	2021/03/04	Create

# Table of Contents

1.	Introduction .....	1
2.	Preliminary preparation.....	2
3.	Apple Pay payment process flow.....	4
4.	Searching order .....	12
5.	Capture and refund.....	15
6.	Searching single credit card order .....	18
7.	Downloading Settlement Report .....	22
Appendix 1.	Checksum.....	24
Appendix 2.	AES Encryption.....	26
Appendix 3.	Error codes.....	31
Appendix 4.	URL Encoding Reference .....	32

## 1. Introduction

ECPay provides merchants who wish to use payment processor that accepts Apple Pay, and supports two main scenarios.

➤ **Pay in apps:**

Enable consumers to pay with Apple Pay in merchant's own app.

➤ **Pay on the web:**

Enable consumers to shop on the Safari, no matter on which device such as iPhone, iPad or Mac that without consumer login.

This API document will detail more about the integration of ECPay's payment services. Before start please make sure to have thorough read of the second chapter—preliminary preparation.

## 2. Preliminary preparation

<p>Ensure that you are eligible for the following criteria before integration</p>	<ol style="list-style-type: none"> <li>2. To become a contracted merchant with ECPay (the application requires about 10-15 working days.)</li> <li>3. <b>Apple accepts only the sale of physical products (in other words, virtual products like game points are not acceptable.)</b></li> <li>4. Merchants must have an Apple's developer account (individual/organization). For organization, the application will require 1 month and the annual fee is USD 100.</li> <li>5. Merchants should design your own payment page and the UI design must comply with Apple Pay's <a href="#">Apple Pay Human Interface Guidelines</a>.</li> <li>6. Apple Pay supports only iOS 9 and above.</li> <li>7. Merchants will have 2 main integration:             <ul style="list-style-type: none"> <li>➢ <a href="#">Payment with Apple Pay</a> (should have SSL).</li> <li>➢ <a href="#">Payment via ECPay</a>.</li> </ul> </li> <li>8. The environment of payment with Apple Pay should support TLS 1.2, including consumer's mobile device and merchant's server.</li> </ol>
<p>Technical preparation before starting integration</p>	<p>Please enroll in as an ECPay's member on Stage (test) server:  <a href="https://member-stage.ecpay.com.tw/MemberReg/MemberRegister?back=N">https://member-stage.ecpay.com.tw/MemberReg/MemberRegister?back=N</a>          When completing the enrollment please send the following data to ECPay's technical support mailbox: <a href="mailto:techsupport@ecpay.com.tw">techsupport@ecpay.com.tw</a>.</p> <ol style="list-style-type: none"> <li>1. <b>MerchantID</b> on ECPay's Stage (it would be a 7-digit numbers).</li> <li>2. The <b>MerchantID</b> on Apply Pay.</li> <li>3. The <b>password</b> to <b>.p12</b> file.</li> <li>4. Apple Pay Payment Processing Certificate: including <b>.p12</b> file and the CSR <b>ECC</b> format. Please compress the two into a zip file).</li> <li>5. The valid date of the Certificate.</li> <li>6. The merchant validation object in <b>Merchant Validation Event: ECC</b> file, which will be obtained via consumer's fingerprint authentication.</li> <li>7. The transaction amount (NTD 5 at least) in <b>Create Payload object</b>.</li> </ol>
<p>Test server of ECPay's admin portal</p>	<p><a href="https://vendor-stage.ecpay.com.tw">https://vendor-stage.ecpay.com.tw</a> (the interface is in Chinese only.)          The admin portal enables merchants to:</p> <ol style="list-style-type: none"> <li>1. Search and check ECPays' order</li> <li>2. Access testing credentials (i.e., <b>MerchantID, HashKey, HashIV</b>).</li> </ol>

### ※Note:

When testing ECPay's API, please confirm the following details:

- (1) ECPay's API support only the HTTPS (443 port). Please use a legitimate domain name system (DNS).
- (2) Please use **HTTP POST** to call ECPay's API.
- (3) When testing API calls (using **FormPOST**), HTML tags (i.e. <br /> <B> <h1>) are not allowed.
- (4) When testing API calls, please do not place credentials into JavaScript, HTML and CSS in order to avoid the theft of credentials.
- (5) Please use **UTF-8** as the format of the API call. ECPay supports only Chinese and English.
- (6) Only NTD is accepted. When testing integrating Apple Pay the **currencyCode** please use **TWD**.
- (7) Be sure to register **MerchantID** and a credential of Apple Pay when testing integration.
- (8) This API does not support Credit card installment.
- (9) To protect customers' rights and online transaction security, ECPay's API service supports only **TLS 1.2** and above.
- (10) If merchants need to accept credit card payment using foreign cards, please enable "accepting foreign credit card" function on ECPay's merchant's admin portal (on **Production** only).

The function of the operating path on portal is: [merchant's admin portal](#) → Credit card receipt → Credit card account settings → Foreign credit card transactions: (please see the following screenshots)

- **Application status:** if **not applied**, please press "application" button.
- **Enabled state:** if **not enabled**, please select "Enable" (the [merchant's admin portal](#) is in Chinese only; the above is translated by Google translation).

**Green industry vendor management background** Contracts negotiated cont

**Account settings**

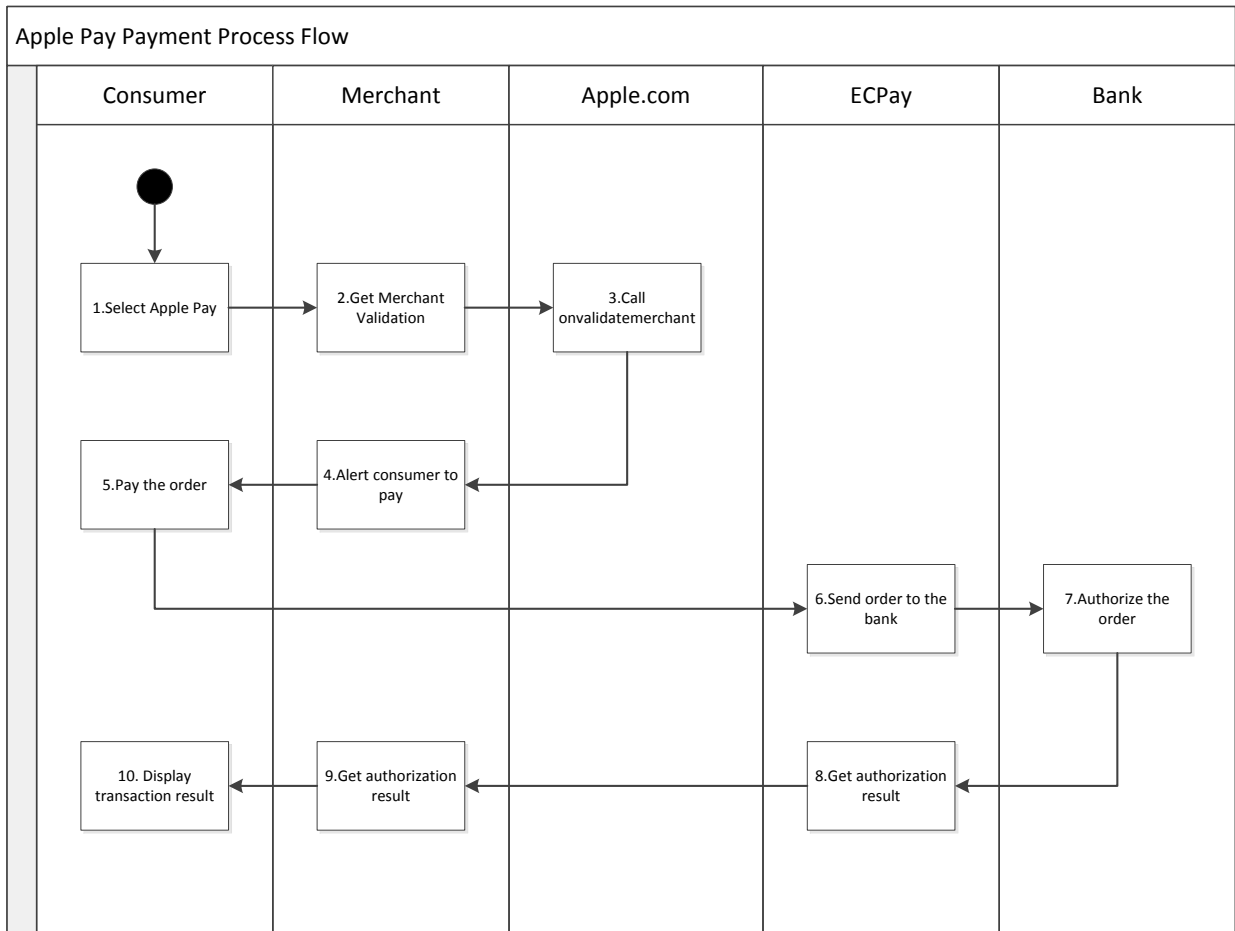
Credit card account information setting

System expiration date	2017/08/26 to 2099/12/31	
Site name	Chiawen	
Credit card authorization completion notification email	<input type="text"/>	
Transaction page (full URL)	<input type="text" value="https://tw.dictionary.search.yahoo.com/"/>	
Behind the scenes back to the program (full URL)	<input type="text"/>	
Foreign credit card transactions	application status	Not applied 14 days after the application for opening a foreign credit card transaction <span style="float: right; border: 1px solid red; padding: 2px;">Application</span>
	Enabled state	<input checked="" type="radio"/> Enable <input type="radio"/> Not enabled
Daily automatic closing	<input type="radio"/> Enable <input checked="" type="radio"/> shut down	
Lock behind-the-scenes verification triggers host IP	<input type="radio"/> Do not use <input type="radio"/> use <input type="text"/>	

(Firstly obtain the verification check code for each transaction, and then send the authorization separated by commas, up to four groups of ip)

- To access ECPay’s credentials (**MerchantID**, **HashKey** and **HashIV**) on Production server, [Please click here to know how to obtain the Production key.](#)
- Merchant's environment:  
Apple Pay supports iPhone 6 and later models (including iPhone SE), iPad Air 2, iPad mini 3 and iPad Pro series models. It also supports Apple Watch series models, as well as Mac and MacBook series models launched after 2012, but Only iPhones and Apple Watches equipped with NFC near-field sensing function can be paid through retail channels, including iPad series and Mac series models, which can only be paid in the App, or use Apple Pay in web services through the Safari browser.

### 3. Apple Pay payment process flow



### 3.2 Creating Apple Pay sessions

- On merchant's web:
  - **ApplePay pay button:**
    - ◆ Design: <https://developer.apple.com/reference/applepayjs/applepaysession#2674254>
    - ◆ Guideline: <https://developer.apple.com/apple-pay/Apple-Pay-Identity-Guidelines.pdf>

#### ApplePay pay button

```
Supports (-webkit-appearance: -apple-pay-button) {
  .apple-pay-button {
    display: inline-block;
    -webkit-appearance: -apple-pay-button;
  }
}

supports not (-webkit-appearance: -apple-pay-button) {
  .apple-pay-button {
    display: inline-block;
    background-size: 100% 60%;
    background-repeat: no-repeat;
    background-position: 50% 50%;
    border-radius: 5px;
    padding: 0px;
    box-sizing: border-box;
    min-width: 200px;
    min-height: 32px;
    max-height: 64px;
  }
}
```

#### ApplePay pay button

```
<div id="divPay">
  <button id="btnApplePay" class="apple-pay-button apple-pay-button-white" lang="tw"
  style="-webkit-appearance: -apple-pay-button; -apple-pay-button-type: buy; width: 400px; height:
  64px;"></button>
</div>
```



### ■ Validating if supporting Apple Pay

- ◆ Function: To validate whether the device supports Apple Pay.
- ◆ Guideline: <https://developer.apple.com/reference/applepayjs/applepayjs/applepayjs/applepaysession#2166537>
- ◆ Scenario: to check if the web browser supports Apple Pay.

```

/* Check whether the web browser supports Apple Pay. */
if (window.ApplePaySession) {
    var merchantIdentifier = 'Your Apple Pay MerchantID'; // Please populate with your Apple Pay Merchant Identifier
    /* Indicates whether the device supports Apple Pay. */
    var promise = ApplePaySession.canMakePaymentsWithActiveCard(merchantIdentifier);
    /* If supports Apple Pay */
    promise.then(function (canMakePayments) {
        if (canMakePayments) {
            /* Displays Apple Pay button */
        }
    });
}

```

### ■ Creating an Apple Pay session

- ◆ Function: To create an Apple Pay session object.
- ◆ Guideline: <https://developer.apple.com/reference/applepayjs/applepayjs/applepayjs/applepaysession#2166537>
- ◆ Scenario: To create a payment request object.

```

/* Set payment request object */
var request = {
    countryCode: 'TW', // The merchant's two-letter ISO 3166 country code.
    currencyCode: 'TWD', // The three-letter ISO 4217 currency code for the payment.
    supportedNetworks: ['visa', 'masterCard'],
    merchantCapabilities: ['supports3DS'],
    total: { label: 'Your Store Name', amount: '100' } // amount of this transaction
};
/* Creating an ApplePaySession */
/* Parameter 1: version, Parameter 2: ApplePaymentRequest object */
var session = new ApplePaySession(2, request);

```

### ■ Getting merchant validation

- ◆ Function: To validate if this is a validated merchant.
- ◆ Scenario: Validation happens when the payment sheet is displayed.

```

/* merchant validation */
session.onvalidatemerchant = function (event) {
    var data = {
        validationURL: event.validationURL
    };
    /* Call your server and pass this validationURL to your server, to request a session from Apple Pay
server.*/
    $.ajax({
        url: 'Your Server URL',
        method: "POST",
        contentType: "application/json; charset=utf-8",
        data: JSON.stringify(data)
    }).then(function (merchantSession) {
        /* to complete the validation for a merchant session */
        session.completeMerchantValidation(JSON.parse(merchantSession));
    });
};

```

### ■ Processing merchant validation on server side (take .NET for example)

- ◆ Function: Parameters setting
- ◆ Scenario: Merchant validation on server side

#### Create merchant validation object

```

string certPath = Server.MapPath("path of merchant certification created of Apple developer account");
string certPwd = "Your merchant certification password";
X509Certificate2 cert = new X509Certificate2(certPath, certPwd, X509KeyStorageFlags.MachineKeySet);

```

#### Create Payload object

```

var payload = new
{
    merchantIdentifier = "Your Apple Pay MerchantID", //Put your Apple Pay Merchant Identifier
    domainName = "Your Website Domain", // Put your website domain(should be same as that of Apple developer
account)
    displayName = "Your Store Name" // Put your store name
}

```

```
};
/* Serialize your Payload*/
string strPayload = new JavaScriptSerializer().Serialize(payload);
```

### Use Transport Layer Security (TLS) 1.2

```
ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12;
```

### Setting of merchant validation request

```
HttpRequest request = (HttpRequest)HttpRequest.Create("the validation URL of the client side");
request.Method = RequestMethod.Http.Post;
request.ContentType = "application/json";
request.ContentLength = strPayload.Length;
request.ClientCertificates.Add(cert); // Add certificates

using (StreamWriter sw = new StreamWriter(request.GetRequestStream()))
{
    sw.Write(strPayload);
    sw.Flush();
    sw.Close();
}
```

#### ■ Payment authorization event

- ◆ Function: Send payment authorization request and get authorization result
- ◆ Scenario: When the merchant validation completes a Touch ID will pop up and then enter payment authorization.

```
/* Payment authorization event */
session.onpaymentauthorized = function (event) {
    var data = {
        payment: JSON.stringify(event.payment),
    };
    /* Send payment object to your server, and the server to handle payment authorization. */
    $.ajax({
        url: ' Your Server URL',
        method: "POST",
        contentType: "application/json; charset=utf-8",
        data: JSON.stringify(data),
        error: function (err) {
            alert(JSON.stringify(err));
        }
    });
}
```

```
    }  
  }).then(function (result) {  
    /* According to the authorization, redirect consumer so show the info of the trade */  
    //Authorization success  
    session.completePayment(JSON.parse(ApplePaySession.STATUS_SUCCESS));  
    // Authorization failure  
    session.completePayment(JSON.parse(ApplePaySession.STATUS_FAILURE));  
  });  
}
```

■ **Processing payment authorization event on server side**

- ◆ Function: To set payment authorization and send to ECPay's API to perform payment authorization.
- ◆ Scenario: After performing Touch ID and then enter payment authorization.

3.3 API integration on ECPay's side

Merchants can call the following API to send payment authorization of ApplePay orders to ECPay and if authorization succeeds ECPay will respond authorization result.

● API URL:

- Production: <https://payment.ecpay.com.tw/ApplePay/CreateServerOrder/V2>
- Stage: <https://payment-stage.ecpay.com.tw/ApplePay/CreateServerOrder/V2>

**Note: When testing this API on ECPay's Stage environment (testing mode), no real authorization will be performed since the bank side does not have authorization of test mode.**

● Merchant request (parameters/fields with red asterisk \* are required):

Parameter	Definition	Type	Definition/Details	Example
*MerchantID	Merchant ID	String (10)		2000132
*MerchantTradeNo	Merchant's order number (order ID)	String (20)	This is a serial number of order/transaction ID created by the merchant, which is a unique identifier combined with upper and lower case alphanumeric characters.	050110030273
*MerchantTradeDate	Time at which the order was created and measured in seconds since Unix epoch	String (20)	The format is yyyy/MM/dd HH:mm:ss	2015/09/21 08:20:18
*TotalAmount	Amount of order	Int	<b>Note:</b> 1. Integer only. 2. New Taiwan Dollar only. 3. The amount restrictions (upper/lower) please check FAQ: <a href="https://www.ecpay.com.tw/CascadeFAQ/CascadeFAQ_Qa?nID=3605">https://www.ecpay.com.tw/CascadeFAQ/CascadeFAQ_Qa?nID=3605</a>	1000
*CurrencyCode	Currency	String (20)	CurrencyCode of Apple's Server to do merchant validation	TWD
*ItemName	Product's name	String (400)	1. If more than 1 product will be populated with this field, each product's name should be separated by hash (#). 2. Maximum length: 400 characters (Chinese and English). ECPay will automatically cut the length if it	Cell phone NT\$20x2#USB Drive NT\$60x1

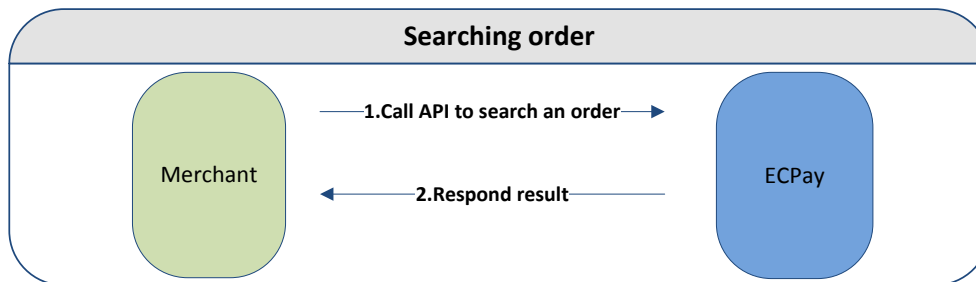
Parameter	Definition	Type	Definition/Details	Example
			exceeds.	
*PlatformID	Platform merchant ID	String (10)	Used by contracted platform merchants.	Since ApplePay is not yet available for platform merchant, please populate this field with [MerchantID].
*TradeDesc	Transaction description	String (200)		
*TradeType	Type of Transaction device	Int	Possible values: 1: In App 2: On the Web	2
*CheckMacValue	Checksum	String	Please see <a href="#">Checksum</a>	
*PaymentToken	Payment token	String	The payment object responded by Apple's server when completing merchant validation. <b>Note:</b> 1. Use AES encryption, please see <a href="#">Checksum</a> . 2. This field does not have to be included in the generation of <a href="#">Checksum</a> .	

● ECPay response:

Parameter	Definition	Type	Definition/Details	Example
RtnCode	Status of authorization	Int	Possible values: 1: success others: failed	Status of authorization: ([a-z,A-Z,0-9])
RtnMsg	Return message	String (200)		SUCCESS
MerchantID	Merchant ID	String (10)		2000132
MerchantTradeNo	Merchant's order number (order ID)	String (20)		050110030273
TradeNo	ECPay's order ID	String (20)	ECPay's order ID Please save it to keep it related to MerchantTradeNo.	1303151740582564
TradeAmt	Transaction amount	Int		
TradeDate	Time at which the order was created	String (14)	The format is: YYYYMMDDHHMMSS	20150921082018
PaymentDate	Time at which the payment was made	String (14)	The format is: YYYYMMDDHHMMSS	20150921082018
CheckMacValue	Checksum	String	Please see <a href="#">Checksum</a>	

## 4. Searching order

- Scenario:  
Merchants request this API to search an order.
- Process:



- API URL:
  - Production: <https://payment.ecpay.com.tw/Cashier/QueryTradeInfo/V2>
  - Stage: <https://payment-stage.ecpay.com.tw/Cashier/QueryTradeInfo/V2>
- Merchants request (red asterisk \* is required):

Parameter	Definition	Type	Definition/Details	Example
PlatformID	Platform merchant ID	String (10)	Used by contracted platform merchants.	
*MerchantID	Merchant ID	String (10)		2000132
*MerchantTradeNo	Merchant's order number (order ID)	String (20)		050110030273
*TimeStamp	Timestamp	Int	<b>Special note:</b> ECPay will verify the timestamp. If ECPay receives the request is more 3 minutes late than the timestamp, the transaction will be failed. Merchants are suggested to synchronize the time of a computer on client- or server-side to another server or nearest reference time source. For more details can be seen <a href="#">here</a> ; another online tool can be found <a href="#">here</a> .	1234567890
*CheckMacValue	Checksum	String	Please see <a href="#">Checksum</a>	

- ECPay response:

Following parameters will be responded to the page with the Content-Type of application/x-www-form-urlencoded (i.e. "key1=value1&key2=value2").

For example: MerchantID=123456789&MerchantTradeNo=123456abc&TradeNo=201203151740582564

Parameter	Definition	Type	Definition/Details	Example
MerchantID	Merchant ID	String (10)		2000132
MerchantTradeNo	Merchant's order number (order ID)	String (20)		123456abc
TradeNo	ECPay's order ID	String (20)		201203151740582564
TradeAmt	Transaction amount	Int		22000
PaymentDate	Date of payment	String (20)	Format: yyyy/MM/dd HH:mm:ss	2012/03/16 12:03:12
PaymentType	Type of payment	String (20)	Fixed value: <b>Credit</b>	Credit
HandlingCharge	Total of service charge	Int		0
PaymentTypeCharge Fee	Retail fee	Int		25
TradeDate	Transaction time	String (20)	Format: yyyy/MM/dd HH:mm:ss	2012/03/15 17:40:58
TradeStatus	Transaction status	String (8)	Possible values: <b>0</b> =order is created but payment is not made. <b>1</b> =order is created and payment is made as well.	0
ItemName	Product name	String (400)		Product 1
CheckMacValue	Checksum	String	Please see <a href="#">Checksum</a>	
gwsr	ID of authorization	Int		10123456
process_date	Date of processing authorization	String (20)	Format: yyyy/MM/dd HH:mm:ss	2013/12/19 13:55:20
auth_code	Code of authorization	String (6)		777777
amount	Transaction/authorization amount	Int		400
stage	Number of installments	Int		3
stast	Amount of down payment	Int		134
staed	Amount of each installments	Int		133
eci	3D (VBV)	Int	This is a value returned from the Directory Server (Visa, MasterCard, and JCB) to indicate the authentication results of cardholder's credit card payment on 3D Secure. Possible values: <b>5, 6, 2, 1</b> : The value returned (ECI) means that transaction was a 3D secure authentication.	5



Parameter	Definition	Type	Definition/Details	Example
card4no	Last 4 digits of credit card	String (4)		2222
card6no	First 6 digits of credit card	String (6)		431195
red_dan	Bonus points deducted	Int		0
red_de_amt	Amount deducted from bonus points	Int		0
red_ok_amt	Actual amount charged	Int		0
red_yet	Remaining bonus points	Int		0
PeriodType	Types of periods	String (1)		D
Frequency	Execution frequency	Int		2
ExecTimes	Number of executions	Int		5
PeriodAmount	Amount of per authorization	Int		200
TotalSuccessTimes	Number of successful authorizations	Int		5
TotalSuccessAmount	Total amount every authorization	Int		1000

## 5. Capture and refund

- Scenario:
 

When the payment is authorized, merchants can call [Searching single order of credit card transaction](#) API to check [[Status](#)] followed by requesting this API to do the following jobs.
- Capture:
  1. If the payment is successful, request [Searching single order of credit card transaction](#) API will respond [authorized]; next merchants can request this API ([Action](#) = C) to capture this order.
  2. When capturing this order, merchants can request [Searching single order of credit card transaction](#) API to check the [[Status](#)]. If it is [To be captured], it indicates this order will be requesting fund to the bank.
  3. If requesting fund process is completed, at this moment requesting [Searching single order of credit card transaction](#) API the [[Status](#)] will be [Captured].
- Refund:
  1. Call [Searching single order of credit card transaction](#) API to check [[Status](#)].
  2. Call this API, and if:
    - [[Status](#)] = [Authorized]: merchants request ([Action](#)=N) (Abandoning Transaction.)
    - [[Status](#)] = [To be captured]:
      - i. To fully refund this payment: firstly request ([Action](#)=E) [Canceling capture] followed by ([Action](#)=N) [Abandoning or cancel transaction].
      - ii. To partially refund: request ([Action](#)=R) [refund payment].
    - [[Status](#)] = [Captured]: request ([Action](#)=R) [refund payment].
- [[Action](#)] instructions:
  - [Action](#)=C: indicating "capture", i.e. to capture this order. It is suggested to capture the order as soon as the payment is authorized.

※Special note:

Orders must be captured within 21 days even if the merchant turned off the automatic capture function on [merchant's admin portal](#) (please see [preliminary preparation](#)).

If orders are not captured within 21 days, an error message-**error\_overDAY** will be showed, and merchants can only contact Green World's customer services to manually capture this order.

If orders are not captured on the 80th day since authorized date, a notification will be sent to the merchant; uncaptured orders over 90 days will be cancelled /abandoned.

- [Action](#)=R: indicating "refund", i.e. merchants can release the credit line captured (held) in the cardholder's credit card account (either in whole or in partial), by revising the amount. The amount to be refunded (released) should not exceed the order amount.

※Special note:

Merchants cannot do refund if merchant's balance in ECPay system is less than the amount to be refunded. In this case, it is advised that merchant deposit or pre-pay enough amount in merchant's account in ECPay's system.

- **Action=E**: indicating "cancelling capture", i.e. merchants can cancel capture and to reverse the order status to previous status (e.g. if **Status** = captured, when requesting **Action=E**, the status will roll back to uncaptured).
- **Action=N**: indicating "abandoning transaction", i.e. merchants abandon (cancel) the transaction/order before the order to be captured. If the order is abandoned, the order will not be processed to the bank for fund.

- API URL:

- Production: <https://payment.ecpay.com.tw/CreditDetail/DoAction>
- Stage: (not available on Stage since Stage cannot perform real authorization.)

- Merchants request (red asterisk \* is required):

Parameter	Definition	Type	Detail	Parameter
*MerchantID	Merchant ID	String (10)		2000132
*MerchantTradeNo	Merchant's order number (order ID)	String (20)		ecpay1234
*TradeNo	ECPay's order ID	String (20)		1234567890
*Action	The action to the order	String (1)	Perform actions to order (details as the above [Action] instructions.) Possible values: <b>C, R, E, N</b> <b>C</b> : capture order <b>R</b> : refund order <b>E</b> : cancelling captured <b>N</b> : abandon transaction	C
*TotalAmount	Total amount of order	Int		22000
*CheckMacValue	Checksum	String	Please see <a href="#">Checksum</a>	
PlatformID	Platform ID	String (10)	Used by contracted platform merchants.	Since ApplePay is not yet available for platform merchant, please populate this field with [MerchantID].

- ECPay response:

Following parameters will be responded to the page with the Content-Type of application/x-www-form-urlencoded (i.e. "key1=value1&key2=value2"). For example:

MerchantID=123456789&MerchantTradeNo=123456abc&TradeNo=201203151740582564

Parameter	Definition	Type	Detail	Parameter
MerchantID	Merchant ID	String (10)		2000132

Parameter	Definition	Type	Detail	Parameter
MerchantTradeNo	Merchant's order number (order ID)	String (20)		ecpay1234
TradeNo	ECPay's order ID	String (20)		201203151740582564
RtnCode	Response code	Int	Possible values: <b>1</b> : success <b>others</b> : failed	1
RtnMsg	Response messages	String (200)		

## 6. Searching single credit card order

- Scenario:

Merchants request this API to search a single order paid by credit card.

- API URL:

- Production: <https://payment.ecPay.com.tw/CreditDetail/QueryTrade/V2>

- Stage: (not available on Stage since Stage cannot perform real authorization.)

- Merchants request (red asterisk \* is required):

Parameter	Definition	Type	Detail	Parameter
*MerchantID	Merchant ID	String (10)		2000132
*CreditRefundId	Credit card authorization ID	Int	The value is same as <a href="#">Gwsr</a>	10123456
*CreditAmount	Transaction amount	Int		
*CreditCheckCode	Check code of credit card transaction	Int	A unique ID of each MerchantID in credit card payment, which can be found at <a href="#">Merchant's Admin Portal</a> .	59997889
*CheckMacValue	Checksum	String	Please see <a href="#">Checksum</a>	

- ECPay's response:

The response will be returned to the webpage in JSON.

For example (following data has be formatted):

```
{
  "RtnMsg": "",
  "RtnValue": {
    "TradeID": "0015625112",
    "amount": "1",
    "clsamt": "1",
    "authtime": "2016/5/12 下午 07:09:17",
    "status": "已關帳",
    "close_data": [
      {
        "sno": "2782343",
        "status": "已關帳",
        "amount": "1",
        "datetime": "2016/5/12 下午 08:00:00"
      }
    ]
  }
}
```

The data responded in JSON have 2 main parts: 1. the main transaction data (in pink background color), and 2.details:

1. Main transaction data:

```

{
  "RtnMsg": "",
  "RtnValue": {
    "TradeID": "0015625112",
    "amount": "100",
    "clsamt": "100",
    "authtime": "2016/5/12 下午 07:09:17",
    "status": "已關帳",
    "close_data": [
      {
        "sno": "2782343",
        "status": "已關帳",
        "amount": "100",
        "datetime": "2016/5/12 下午 08:00:00"
      }
    ]
  }
}

```

Main data:

Parameter	Definition	Type	Definition/Details	Example
RtnMsg	Response message	String (200)	Possible values: <b>ok</b> : success <b>error Stop</b> : no merchant matched or the merchant has expired. <b>error nopay</b> : no TradeID matched. <b>error</b> : data checked failed.	ok
RtnValue	Response value			
TradeID	ID of authorization	Int		0015625112
amount	Transaction amount	Int		100
clsamt	Amount captured	Int		100
authtime	Time of authorization	String (24)		2016/5/12 下午 07:09:17
status	Transaction status	String (30)	<p>If there is no data of capture, possible values are (in Chinese only):</p> <p><u>已取消</u>(Cancelled): Transaction has been cancelled.</p> <p><u>未授權</u>(Unauthorized): Authorization is not completed by the bank.</p> <p><u>已授權</u>(Authorized): authorization is completed.</p> <p>If there is data of capture, possible values are (in Chinese only):</p> <p>銀行拒絕 (Refused by bank)</p> <p>要關帳 (To be captured)</p> <p>關帳中 (Capture in process)</p> <p>已關帳 (Captured)</p> <p>要取消 (To be cancelled)</p> <p>取消中 (Cancellation in process)</p> <p>已取消 (Cancelled)</p> <p>銀行追回中 (Bank chargeback in progress)</p> <p>銀行已追回 (Bank chargeback complete)</p> <p>批次失敗 (Batch failed)</p> <p>不明 (unknown cause)</p> <p>操作取消 (Operation cancelled)</p>	已授權 (Authorized)

## 2.Details

```

{
  "RtnMsg": "",
  "RtnValue": {
    "TradeID": "0015625112",
    "amount": "1",
    "clsamt": "1",
    "authtime": "2016/5/12 下午 07:09:17",
    "status": "已關帳",
    "close_data": [
      {
        "sno": "2782343",
        "status": "已關帳",
        "amount": "100",
        "datetime": "2016/5/12 下午 08:00:00"
      }
    ]
  }
}
    
```

## Details:

Parameter	Definition	Type	Definition/Details	Example
sno	ID of capture	String (20)	Only when the order has been captured, this parameter will be returned.	2782343
status	Status of credit card order	String (30)	Possible values are (in Chinese only): 銀行拒絕 (Refused by bank) 要關帳 (To be captured) 關帳中 (Capture in process) 已關帳 (Captured) 要取消 (To be cancelled) 取消中 (Cancellation in process) 已取消 (Cancelled) 銀行追回中 (Bank chargeback in progress) 銀行已追回 (Bank chargeback complete) 批次失敗 (Batch failed) 不明 (unknown cause) 操作取消 (Operation cancelled)	已關帳 (Captured)
amount	Transaction amount	Int		100
datetime	Time of capture	String (24)		2016/5/12 下午 08:00:00



## 7. Downloading Settlement Report

- Scenario:

Merchants can download their credit card statement report in CSV format.

Step 1. Merchant: call ECPay's API by sending the following data.

Step 2. ECPay: responds a CSV file for the merchant to download.

- API URL:

- Production: <https://payment.ecPay.com.tw/CreditDetail/FundingReconDetail>

- Stage: (not available on Stage since Stage cannot perform real authorization).

- Merchants request (red asterisk \* is required):

Parameter	Definition	Type	Definition/Details	Example
* MerchantID	Merchant ID	String (10)		123456789
*PayDateType	Type of date to be queried	String (10)	Type of date to be queried Possible values: <b>fund</b> : date of ECPay deposit the fund into merchant's account <b>close</b> : date of capture <b>enter</b> : date of payment into merchants' account	close
*StartDate	Start of the date range to be queried	String (10)	Format: yyyy-MM-dd	2015-02-12
*EndDate	End of the date range to be queried	String (10)	Format: yyyy-MM-dd	2015-02-12
CharSet	Meta charset	String (1)	Meta charset of format to be downloaded. Possible values: <b>1</b> : Big5 <b>2</b> : UTF8 If this parameter was not sent or null value, the default would be Server Default format.	1
*CheckMacValue	Checksum	String	Please see <a href="#">Checksum</a>	

- ECPay's response (columns in the CSV, in Chinese only):

Column	Type	Definition/Details
ID of authorization	String	
Authorization code	String	
ID of capture	String	
MerchantTradeNo	String	
Transaction date	String	
Date when ECPay ask bank for fund	String	
Transaction amount	Int	
Service charge	Int	
%	Float	

---

Column	Type	Definition/Details
Amount of fund to be deposit into merchant's account	Int	

## Appendix 1. Checksum

- When sending data to ECPay's API, all of the key and value must go through checksum except for [CheckMacValue] and [PaymentToken]. Following demonstrates how the checksum works.

- Sort alphabetically (A-Z) and link with an ampersand (&) to the request body (with Content-Type of application/x-www-form-urlencoded).

```
currencyCode=TWD&ItemName=手機 20 元 X2#隨身碟 60 元
X1&MerchantID=2000132&MerchantTradeDate=2017/03/21
17:02:00&MerchantTradeNo=20170321170200889&PlatformID=&TotalAmount=100&Trade
Desc=ecpay 商城購物
```

- Sandwichwith HashKey and HashIV

```
HashKey=5294y06JbISpM5x9&currencyCode=TWD&ItemName=手機 20 元 X2#隨身碟 60 元
X1&MerchantID=2000132&MerchantTradeDate=2017/03/21
17:02:00&MerchantTradeNo=20170321170200889&PlatformID=&TotalAmount=100&Trade
Desc=ecpay 商城購物&HashIV=v77hoKGq4kwxNNIS
```

- Urlencode the string:

```
HashKey%3d5294y06JbISpM5x9%26currencyCode%3dTWD%26ItemName%3d%e6%89%8b%
%e6%a9%9f20%e5%85%83X2%23%e9%9a%a8%e8%ba%ab%e7%a2%9f60%e5%85%83X1%
26MerchantID%3d2000132%26MerchantTradeDate%3d2017%2f03%2f21+17%3a02%3a00%
26MerchantTradeNo%3d20170321170200889%26PlatformID%3d%26TotalAmount%3d100%
26TradeDesc%3decpay%e5%95%86%e5%9f%8e%e8%b3%bc%e7%89%a9%26HashIV%3dv7
7hoKGq4kwxNNIS
```

- Lowercase the string:

```
hashkey%3d5294y06jbispm5x9%26currencycode%3dtwd%26itemname%3d%e6%89%8b%e
6%a9%9f20%e5%85%83x2%23%e9%9a%a8%e8%ba%ab%e7%a2%9f60%e5%85%83x1%26
merchantid%3d2000132%26merchanttradedate%3d2017%2f03%2f21+17%3a02%3a00%26
merchanttradeno%3d20170321170200889%26platformid%3d%26totalamount%3d100%26t
radedesc%3decpay%e5%95%86%e5%9f%8e%e8%b3%bc%e7%89%a9%26hashiv%3dv77ho
kgq4kwxnnis
```

## v. Using SHA256 to encrypt the string:

```
BDC2A456448FDB2F1A14C6098C79E9326D5DB39342630AEB52E30AFB64DC4A82
```

**Special note:**

If converting urlencode from PHP to C#, please follow the rule of [URL ENCODING REFERENCE](#). For example, in PHP, ! would be converted to %21, which is not suitable in C#. For PHP developer please use `str_replace` function to convert %21 back to !. Following are example of converting function in PHP.

```
$sMacValue = str_replace('%2d', '-', $sMacValue);  
$sMacValue = str_replace('%5f', '_', $sMacValue);  
$sMacValue = str_replace('%2e', '.', $sMacValue);  
$sMacValue = str_replace('%21', '!', $sMacValue);  
$sMacValue = str_replace('%2a', '*', $sMacValue);  
$sMacValue = str_replace('%28', '(', $sMacValue);  
$sMacValue = str_replace('%29', ')', $sMacValue);  
$sMacValue = str_replace('%20', '+', $sMacValue);
```

For other coding language please follow its converting rule.



```

MuY3JsMA4GA1UdDwEB/wQEAwIBBjAQBgoqhkiG92NkBglOBAIFADAKBggqhkiOPQDQAgNnADBkAjA6z3KDURaZsYb7NcNWY
mK/9Bft2Q91TaK0vGcgV5Ct4n4mPebWZ+Y1UENj53pww4CMDIt1UQhsKMFd2xd8zg7kGf9F3wslW2WT8ZyaYISb1T4en0bmc
ubCYkhYQaZDwmSHQAAMYIBYDCCAIVwCAQEwGYYweJEUmCwGA1UEAwWlQXBwBwGUGQXBwBwGijYXRpb24gSW50ZWdyYXRpb
24gQ0EgLSBHMzEmMCQGA1UECwwdQXBwBwGUGQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkxZzARBGNVBAoMCKFwcGllEluYy4
xCzAJBgNVBAYTAiVTAghoYPAz2cynDzANBgIghkgBZQMEAgEFAKBpMBGCSqGSIb3DQEJAzELBgqhkiG9w0BBwEwHAYJKoZIh
v cNAQkFMQ8XDTE2MTIwODA4NTQxMFowLWYJKoZIhvcNAQkEMSIEICAG72ATE79/cRRC8cpAO0+MIW3+fi8Vl6EtmkYDAJMA
oGCCqGSM49BAMCBEGwRglhAlZAUADfteo1Pb9+YTaVR0Sm4HmjCRf1587692RZoy0xAiEA2BPHpVID4zCKVzS9eCCeUpwl+Rf
9yr8iTMGSSceN/OAAAAAAAAA=",
    "header":
    {
        "publicKeyHash": "xLBxijBsfyoaFVUIEUEQptPsmD4Wlt491ovV5DKNag=",
        "ephemeralPublicKey": "MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAE/WwsbwNtdjBmX94iGQh5Z4Cyt7v7rMNAO2Hgi9
3HUUpSSJ5pl+5TYM4UtdIWfDbBzXQiwCAKT3jreo1cLPAdzg==",
        "transactionId": "85a158652b0d98379cb817dd53e97dfd8131dde2f578128482b6dee4c9e98eb9"
    },
    "version": "EC_v1"
},
    "transactionIdentifier": "85A158652B0D98379CB817DD53E97DFD8131DDE2F578128482B6DEE4C9E98EB9",
    "paymentMethod":
    {
        "network": "Visa",
        "type": "debit",
        "displayName": "Visa 0492"
    }
}
}

```

- AES Encryption(with 128 bits, CipherMode: CBC and PaddingMode:PKCS7):

Encrypt the JSON string with AES encryption and then encoding with Base64.

```

0bOrCfx8Vz4YGruVYMH7EKaF62z0l0uefvb23Ev8sZ1oRWGyRCFJZ1J7QXvdVcU32Vg4f55aR1KaFwfEs9I5kxUnO9G+WHXw
9EkGo+hypnxUUB7ydUyM8elapNBc8PrZTCTD0RXYuuey9dwoBia6OH1I9lqsFKCedCUI7EYotRGQg21Bi5QQarsO2hatiXbCj
MqeMTsLXEGNJPnZdYvSJANJLQOBK+kOfCqGMFruYj7ajR9OdZ4kqk+vJtF1etzG2Yt6jKJ6Z9pRkP/VILx8Cdb6nNx97VZOJM1S
dYGSfBxX1AqWoydm+wOpmx26jKUJqGSFPLFsYVbJwqAySju+z4pLJkv4Q2WMKnyhxaEfCUyONLnUauLLec6ljddulSgw5cZtK

```

MmcpteKUWriefzmVZ/PCm3g/zSYSTtbrxVjJXFYMQWJ36FtwnhVRV3vStSfLWxBb6cBw2ggFpMHgYo3Pr01pWYYQR/CM  
aTx78z2StpcddNBU11rq7D8ctZBUUMRhpPIE4hr7kt/HawzsZQvq2S7Jh3lsY237+vScyE3LiFIEPR1NuYVUmMvmaUisAsl7ym  
3luVJ99xhIVsLaX/TG/tuQ5dd8nsKY9XnWh3Sc5Z+n7faqgFHuVgDyNlv0404iirBmubmETQ6BDWX2FzvBbJymWaXyuhBAxV  
KY5Ifb4i9djScRrzDcD1fGz40AZsK/bKKmAYvlhWAoVK/M0XMQGMs1yRPGUCtZ5/qFgfZHBS9E8pv30StkPTQdrjertv2UHK6  
EV+goF5JD7K0gfMFtYrgWZmdekmtOKj/BeMpBF6FE2PDsh19b5kd4jQAtNJ0jThujWu9IQ6xVtHrvMPZWsytt1q7ykYmt8p  
m+bxSI5Al+VlopiYjdMaPGSQ7TEryCBtnqiESpfGCXZsazmxWScox8fF//cE1Z8nMpLiCPX6tJJ6ZS6Cgu/Zl0EzJbZXSUHOzPYartt  
Vfc+nJFpfdgkPtn06kE6I+2t+ZRuwgCA70s7WNSH8cDX5gNc3XHVTfc8sRTpVR9nLvUwyqUoY6I6uUcE8AJsfqGSI795sl688j6P  
06l70RqA2RvqHuDaBOegIQ56ll7z9tdmg70MB9iCfLidGihMkUTA1dqWOyOzNVVjC/EQHCoWJ9x6cZxLTHX2xqMt91O8n3Jy  
7gu+ULP6vqD+r5b9wZ8iSxj483GyAg6m+WR4tPS55+SAI2gwL/6o0FEyMYKeyTZ8OuEYSh69207av7xEJMFxIhWUgJpFC8du  
ZjorL3A77PNeAxwhCMhUYyHqUTVEi+FTwHIpVWL2a1PQ4FuCpFFRyLwkk8lqCUNUeO0l+arVBmWoGXX+rkGUqmre6kJOlj  
HeZcEYFDbDv6Dq2PPF0v8iAeo/moxOyeg3PuCfieOAC5wfF0GrKUnzXyE8nNV/Z9dMgDf8J/Z6nTP2x+mDnCCIMpc5QSe9T  
Bmxec+SGRNJU7P2mKLWekxlaicWqHlfwF44N15SjdB1YTTpROsaxZBLSeYcVS7Cg+UiEGnUWX+r69BO4oNWBQVZ+CYG2Ct  
46xlyj31kST1HvrNkbdjVp7eLzUTGf6ZzoDwlwH+QBf0tOISnk+DGB/Z6mAHPreAd/Sbtg5hTz2IE3J9dqTMU8fqoOHZaMKf4e  
iSYwT0Uxsrs07JjUv0xuqPrs4G7i9fdrvrXjVYX1S0OIVCwkBo4jFfXoP3PsdXp/dp+ocTHUTYrf36gBI7ZnJpzfNA7UsODNNGNe6  
xnzZyfsMQKcMeN5aHYuaS1PUTFktN1hzfSbVLPFhN2CslMQFOzma3Ss+pt4ajk7cSB8/h1S71DDUjS3YQLot0qEehsNb0qD  
T4lfdc9dfTkjPqs4UI5Pu5ClgVa47aev/YxJy3RMKcd7T4xOT/c27/iKiYs6swW1aAraTJVCe1YMhjIN1QsCcw4lky9t6A/lZfNef6  
TVIU7q3vqI7sh8b/ilCgU7YdNVkmenv5gPqANGYeNg9rkC0vrhIDM3eIXIWQuS1SjF9CU3RtfVlb8ByiQqKyMPMEHlpLs3o+zK  
B+xpGsc5iXdDdUQMn/pmn8gKq6P174NSMHXZqITPa6wStZyI4TwW9rgYHsFfsgqau3Y+5T+AQUJaWwEukF4gv2BRXQ9p3ff  
ImT1gCopuRwfeo1mp6DofZPbWWFnhPYRZlt4NlLebD11QnWVVKDO4WwaDHMK2ocap+/mmYoR+65Inat7HhoYJNeNENF  
BKe4yRzxdF+hq8LRR9Ya4J3pWSliAzHcXXJAKmvGFzUs5cSQvAvBwN/8YLSepEpuFFssRDjqv2XdEDGVQsP6Sy2ix4hQqQ/TTj  
IACTlvZ4H8kkUNc2wEschA9nyBNOemTNeZ58ubStnQa9tHe32ztluJvKhlJzU4mEiEzStmHsDhqmChPOvnNsw/8cuy7xgDXez  
5Whs3x7KLJvF8fA/g8aGIW3YQc5o1lktPLlxqjjXvohBslvgCs8OR0DNjvYz+YG+WYzfe3zSlp63XDxEsh1EtRIA9Z60BSWm/F  
WYB/4vnBhRQa0TwtFmFmHNoSljdljprVebSvH0Tjx5UyNEVRHntDc1UlM5zISotOk7ar5t81vJSx5lgWfRmMaHyBuob47w8  
2BWbBVzfTayCyOXrjUcsGa5Snovd4k9h/tDNVIK5IH6mqQ+MwU9ID7YvYH1/ftbmdsqGxnnDPxiTKAT0owKXboSSzdN+/e  
/HkMCsG3pzRPeVuRwsvYH+rKQiQwomeszAG3DocU6qYxjmNC7hbqgljOyD4yyKlrvH5MWSNcP8G9nXPeArt8qx3lx7xViB  
MpuaofwdU6pypG4Rn6KJFYViFvbpOCIKP7/mPzih6+nm7ET/HvnWbkl2LupxzUsZUiiC09etBTXxFWGcF3RpayiSWy7WGQ  
M7cn1FjXmbw3iVo8xUsnDVfoZ+j7j4IICCGDdZSKQmCsgfofo/dc/kPK/YMNI8N6wbp9yXhXtdg3gCzR/SdzZ1UrgYsGrpKlgO  
EhSxWfi/GRwRREWzo0+eqOMGEwmGggVMDVktbNorT2WvHDEUBpj5mVw/unCcYRLLZ0ZX8YNByJ/A5c1Ca+Ze+5AilKz  
ZEI6svXm2qFHktaA8gqQTrYhKUb6ZypIPFX2SENcrgMjopCit7JnxzLfrwV6gZliMpUI+fSl4RaQOpAjicWHh2XnSUQ6qr12mC  
fyvqoWQmTk4AEDEqU5bRd6P4AXt1+EMVksIEJMGUOEVO/hYaR3YiHwycL6fbkGuN1YKS9H7RQhKEISDbpcXecp7B08gK0T  
PndULI6Vf7e6/madLfdwilNDJz3W7IL5m48M7nGpqqem+8BPEkmtKc+5fbz8XxHKxMQhe8A4i611ORYQdnWPuRhfdjdbkIf7  
pLAPXJKc/bivM2a6QPvjUkEibii/o7c8Gm4zlX09CyKhdw5pb/uS9GyyRio1NcN/woyFagolbw+Z/L5W2m4kLxNunQkoYUhf  
anlcWrLnllepq32FkbWlaL+VWJs04iINHgJlwr2mBRigypu/aZuU2fUQTycPw9b62wAJFRLbJNMl1vrurvMm5bXen3Ogsz0XKg4j  
ex/woUBnBwEFleKEteLTrzfgOLEGQP4inRkf5tiOqcHG5NDRz7dJMqII1PLyGdkdB1luYmlwRyBwtiwqyN96gSDxgpQzi2Udb  
4JvwMMt7s8owDNxqJKMUZCeTufXyguPU3gwz5pUuCDUlyCTIsOZ4MFRtTydrT+PZJOMnAqxwF0e+86OVYgLTm37PqCW2Cn  
PQ8yb4BxsGqzY+iRb/DEU7bAdfUKKr6VkjU0ow1Wjm8SV1R1Ejfk6QNHXfp2sQ+ulzv7BR8Y1FussRAUwdncigyf7CoJsjd27L  
HP6CYT6GGk9Z7Qf5N/ork4zAtwLbUrZlM8fozz0uzVF6+IEBSIUyZ/Fta70sqcF6vuaxgJ4fyM/iAfRHMqWpFjztZ+/rAIPU8IYJ  
Y9kc6DF6axTCFvP0j2RaFKH7jH5Ep3dK7yJaf7LaChsG5Jm0nnZaUDQWRwX5W96tSZlCfUv7rkgil2PQUtIYj+ETmw38aGD5

RaUi8C/bA9queqwZ5O8gAUmF5oENXUSGiYmjUYeugNoX8dyg6xGjbNidoy/PRDktXiLXxuk2vDsUqVlJddotH5HqZ9Sf735LD  
aA6Ko+NzWJnffUMNCEKLPwsk0ia7604Uq9vOXO8kiWuusikTI1V9KrdY/fiU7xqkQ8JrT7v0Pf7aHPlmu9pB7pObUqN0jp/BA8  
FXxEVpKIKL+IdkaSgW42CK9C08AnG1Kyj91bC5h+CF8nPNDgC8uZ6n1wCBOiKy3QsmXTZi7KYbMvHWvenxBvVup+U+JQMP  
TA+d+pF2zpxdNI9w2G+F7Wp/1zAyL54n3IANR2s6Xo7sHZ8kLzaUNSQSUVleO0p0NEZQ4ahFx+eF0I2VmyNEEdEhTo91nX4  
KQSYTVUnn9+mR/+WOObFF1oQ2d4k8Fv/X7QD6ZhdSZzTii3milVoDFBUk8syuLY04komsObTrvIkoZGLWBkzW7WlnhxiFxu  
YqP1S/SmUAq1ZSKO/E0q6q5mxtLmK1ggjqQC1bk4NM8Zmm4HtWtjRq4pADpON/Npa+073r2f/IRLcBkMzn6vgW9vH3Nq  
qsEr0JHKhY43SEnIO9tBQhAffaQnhbMsTriw/aFcJGnHhalLAJ+sKUGUlsMRMgVXAA0RwoBOyMLwu4du5vVrVBPnw8UurcT  
i19toqn1UribrbeCuuG6xd8giNNRdSelK9B9srS3wImcySrzxK5eqmnuhbDeHg0+e8XMfL8FSfFNWJTOlqRWfA8S1xCcn3SfW  
pjXu41d9ZGUFU74Y/5kJUkgtO22cBxyujRQvRqj9qeW2+bg8INR8lqtCZAW6huO3Wk3dwqLrTzIo2S7mlUGyKpCCnAZ0ZSA  
vMHO/PgXpOBEc6mskYx2CNCJfKnydKW8U+xVN1m/Ua+tkMh0b9MfeHG8uPXmmcs7XVC1WYm2ApK6/Z93eEEnFvPgNtF  
exTYWwg1AArOvg+VzeWyyUJ9UpR91DmAAHJGp/ih/HpLpihTr5Plg7I3Gu3i97pmmKF+C4JKVckpDZ3kp0T2SBuPewNzb  
OuBt4fySvldail4dEJfeVEekc1QjSmn4/KuB9mSI=

UrlEncode the string encrypted of the above:

0bOrCfx8Vz4YGruVYMH7EKaF62z0l0uefvb23Ev8sZ1oRWGyRCFJZ1J7QXvdVcU32Vg4f55aR1KaFwfEs9I5kxUnO9G%2bWH  
Xw9EkGo%2bhypnxUUB7ydUyM8elapNBc8PrZTCTDORXyueey9dwoBia6OH1l9lqsFKCedCUI7EyotRGQg21Bi5QQarsO2hat  
iXbCjMqeMTsLXEgNJPnZdYvSJANJLQOBK%2bKOfCqGMFruYj7ajR9OdZ4kqk%2bvJtF1etzG2Yt6jKJ6Z9pRkP%2fVILx8Cdb6n  
Nx97VZOJM1SdYGSfBxX1AqWoydm%2bwOpmx26jKUJqGSFPLFsYVbJwqAySju%2bz4pLJkv4Q2WMMknyhaEfCUyOLnUau  
LLec6lJdduISgW5cZtKMmcpTEKUWriefzmVZ%2fPCm3g%2fzSYSTtbrxVjJFYMQWJ36fFtwnhVRV3vStSflWxBb6cBw2ggFp  
MHgYo3Pr01pWYYQR%2fCmaTx78z2StpcddNBU11rq7D8ctZBUUMRhpPIE4hr7kt%2fHawzszQvq2S7Jh3lsY237%2bvScyE  
3LiFIEPR1NuYVUmMvmaUisAsi7ym3luVJ99xhlVsLaX%2fTG%2ftuQ5dd8nsKY9XnWh3Sc5Z%2bn7faqgFHuVgDyNiv0404iIR  
bMubmETQ6BDWX2FvzBbJymWaXyuhBAxVKY5Ifb4i9dJScRrzDcD1fGz40AZsK%2fbKkmAYvIhWAoVK%2fM0XMQGMs1yR  
PGUCtZ5%2fqFgzHBS9E8pv30StkPTQdrcjertv2UHK6EV%2bgoF5JD7K0gfMftYrgWZmdekmtokj%2fBeMpBF6FE2PDsh1  
9b5kd4jQAtNJ0jThujWu9IQ6xVtHrvMPZWsytt1q7ykYmt8pm%2bbxSI5Al%2bVlopiYjdMaPGSQ7TEryCBtnqIESpfGCXZsaz  
mxWScox8fF%2f%2fcE1Z8nMpLiCPX6tJJ6ZS6Cgu%2fZl0EzJbZXSUHOzPYarttVfc%2bnJFpfdgkPtn06kE6I%2b2t%2bZRuwgC  
A70s7WNSH8cDX5gNc3XHVTfc8sRtpVR9nLvUwyqUoY6I6uUcE8AJsfqGSI795sl688j6P06I70RqA2RvqHuDaBOeglQ56ll7z9t  
dmg70MB9iCfLidGihMkUTA1dqWOyOzNVVjC%2fEQHCoWJ9x6cZxlTHX2xqMt91O8n3Jy7gu%2bULP6vqD%2br5b9wZ8iSx  
j483GyAg6m%2bWR4tPS55%2bSAI2gwL%2f6o0FEyMYKeyTZ8OuEYSh69207av7xEJMFxIhWUgJpFC8duZjorL3A77PNeAX  
whCMhUYyHqUTVEi%2bFTwHlpVWL2a1PQ4FuCpFFRyLwkk8lqCUNUeO0l%2barVBmWoGXK%2brkGUqmre6kJOlJHeZcEY  
FDbDv6Dq2PPF0v8iAeo%2fmoxyOyeg3PuCfieiOAC5wfF0GrKUnzXyE8nNV%2fZ9dMgDf8J%2fZ6nTP2x%2b2mDnCCIMpc5Q  
Se9TBmxc%2bSGRNjU7P2mKLWekxlaicWqHlfwF44N15SjdB1YTTpROSaxZBLSeYcVS7Cg%2bUiEGnUWX%2br69BO4oNW  
BQVZ%2bCYG2Ct46xljy31kST1HvrNkbdjVp7eLzUTGf6ZzoDwlwH%2bQBf0tOISnk%2bDGB%2fZ6mAHPreAd%2fSbtg5hTz2  
IEJ9dqTMU8fQOHZaMKf4eiSYwT0Uxsr07JjUv0xuqPrs4G7i9fdrvrXjVYX1S0OIVCwkBo4jFfXoP3PsdXP%2fdp%2bocTHUT  
Yrf36gBI7ZnJpzfNA7UsODNNGNe6xnxZyfsMQKcMeN5aHYuaS1PUTFktN1hzfSbVLPfHn2CsIMQFOzmA3Ss%2bpt4ajk7cSB  
8%2fh1S71DDUjS3YQLot0qEehsNb0qDT4lfd9dfTkpQs4UI5Pu5ClgVa47aev%2fYxJy3RMKcd7T4xOT%2fc27%2fKiYs6s  
wW1aAraTJVCe1YMHjIN1QsCcw4lky9t6A%2fZfNEf6TVIU7q3vql7sh8b%2filCgU7YdNVkmenv5gPqANGYeNg9rkC0vrhIDM  
3eIXIWQuS1SjF9CU3RtfVib8ByiQqKyMPMEHlPs3o%2bzKB%2bXpGsc5iXdDduQMn%2fpmn8gKq6P174NSMHXZqITPa6w  
STyI4TwW9rgYHsFfsgqau3Y%2b5T%2bAQUJaWwEukF4gv2BRXQ9p3ffImT1gCopuRwfeo1mp6DofZPbWWFnhPYRZlt4NLI



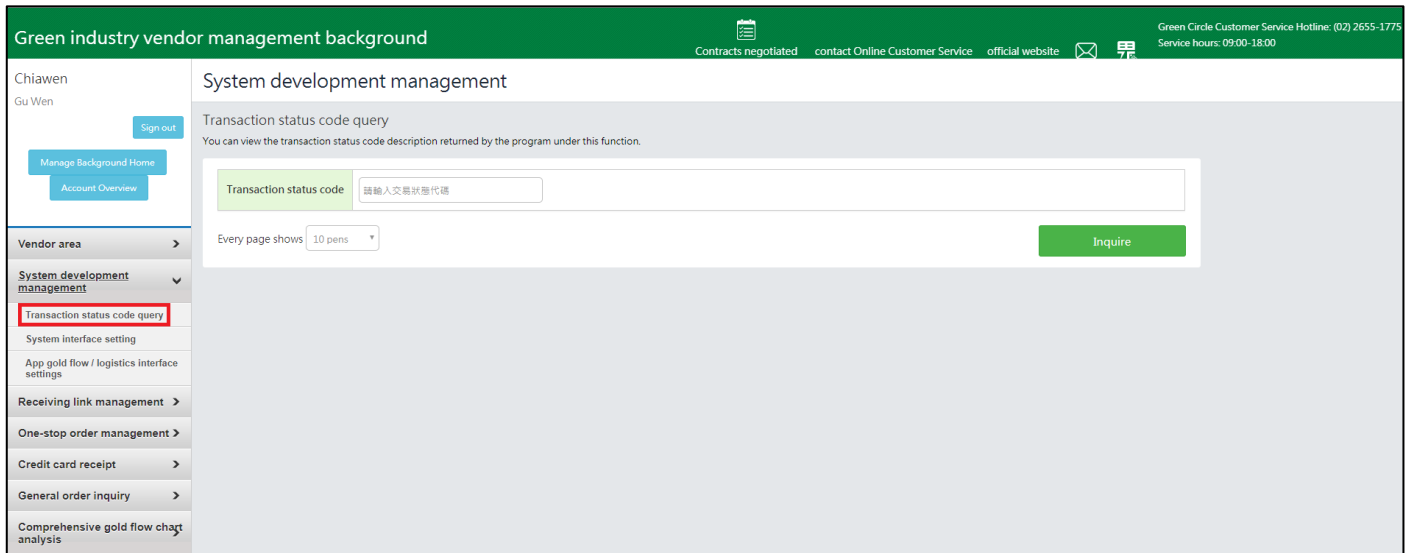
ebD11QnweVkdO4WwaDHMK2ocap%2b%2fmmYoR%2b65Inat7HhoYJNeNENFBKe4yRzxdF%2bhq8ILRR9Ya4J3pWSliAzH  
cXXJAkmvGFzUs5cSQvAvBwN%2f8YLsEpEpuFFsRDjqv2XdEDGVQsP6Sy2ix4hQqQ%2fTTjIACTlvZ4H8kkUNC2wEschA9nyB  
NOemTNeZ58ubSTnQa9tHe32ztIujVkJzU4mEiEzStmHsDhqmChPOvnNsw%2f8cuy7xgDXez5Whs3x7KLJvF8fA%2fg8aGI  
W3YQc5o1lktPLlxqjXvohBslvgCs8OR0DNgjvVzY%2byG%2bWYzfe3zSlp63XDxESH1EtRIA9Z60BSWm%2fFWYB%2f4vnBhR  
Qa0TwTfMfMhNoSljdIjIprVebSvH0Tjx5UyNEVRHntDc1UlM5zISotOk7ar5t81vJSx5lgWfRmMaHyBuob47w82BWBvzFtAy  
CyOXrjUcsga5Snovd4k9h%2ftDNVIK5hIH6mqQ%2bMwU9ID7YvYH1%2fftBmdsqGxnnDPxITkAT0owKXboSSzqdN%2b%2f  
e%2fHkMCsG3pzRPeVuRwsYH%2brKQiQwomeszAG3DocU6qYxjmNC7hbqgljOyD4yyKlrvH5MWSNcP8G9nXPeArt8qx3I  
7xViBMpuaoFwdU6pypG4Rn6KJFYViFvbpOCIKP7%2fmPzih6%2bnm7ET%2fHvnWbIkI2LupxzUsZUiiC09etBTXxFWGcF3Rp  
ayiSWy7WGM7cn1IFjXmbw3iVo8xUsnDVfoZ%2bj7j4IICCGDdZSKQmCsgofo%2fdc%2fkPK%2fYMNi8N6wbp9yJxHtdg3g  
CzR%2fSdzZ1UrgYsGrIpKlgOEhSxWfi%2fGRwRTEwzo0%2beqOMGEwmGggVMDvkTbNorT2VwHDEUBpj5mVw%2funCc  
YRLLZ0Zx8YnByJ%2fA5c1Ca%2bZe%2b5AilKzZEI6svXm2qFHktaA8gqQTrYhKUB6ZypIPFX2SENcrgMjopCit7JnxzLfrwV6g  
ZliMpUI%2bfSI4RaQOpAjiCWHh2XnSUQ6qr12mCfyvqoWqMtk4AEDEqU5bRd6P4AXt1%2bEMVksIEJMGUOEVO%2fhYaR  
3YiHwycL6fbkGuN1YKS9H7RQhKEISDbpcXecp7B08gK0TPndULI6Vf7e6%2fmadLfdwilNDJz3W7IL5m48M7nGpqqem%2b8  
BPEkmTkc%2b5fbz8XxHKxMQhe8A4i611ORYQdnWPuRhfdjBklf7pLAPXJKc%2fbivM2a6QPvjUkEibii%2fo7c8Gm4zlX09CyK  
hdw5pb%2fuS9GyyRio1NcN%2fwoyFagolbw%2bZ%2fL5W2m4kLxNunQoYU%2ffdanicWrLNlepq32FkbWlAL%2bVWJsJ0  
4iIHgJlwr2mBRigypu%2faZuU2fUQTycPw9b62wAJFRLbJNMI1vruvMm5bxEn3Ogsz0XKg4jex%2fwoUBnBwEFleKEteLTrzZ  
fgOLEGQP4inRkf5tiOqCHG5NDRz7DJMQlI1PlyGdkdB1luYmlwRyBwtiwqyN96gSDxgpQzi2Udb4JvwMMt7s8owDNxqJKMU  
ZCeTufXygPU3gWz5pUuCDUlyCTIsOZ4MFrTtydrt%2bPZJOMnAqxwF0e%2b86OVYgLtM37PqCW2CnPQ8yb4BxsGqzY%2bi  
Rb%2fDEU7bAdfUKKr6VkJU0ow1Wjm8SV1R1EjfK6QNHXfp2sQ%2bulzv7BR8Y1FussRAUwdncigfy7CoJsjd27LHP6CYT6GG  
k9Z7Qf5N%2fork4zAtwLbUrZIm8fozz0uzVF6%2bIEBSIUyZ%2fFta7OsqcF6vuaxgJ4fyM%2fiAfrHMqWpFJztZ%2b%2frAIP  
U8IYJ9k6DF6axTCFvP0j2RaFKH7jH5Ep3dK7yJaf7LaChsG5Jm0nnZaUDQWRwX5W96tSZlCfUv7rkgil2PQUtlYj%2bETmw  
38aGD5RaUi8C%2fbA9queqWZ5O8gAUmF5oENXUSGiYmjUYeugNoX8dyg6xGjbNidoy%2fPRDktXiLXxuk2vDsUqIvljddotH5  
HqZ9Sf735LdaA6Ko%2bNzWJnffUMNCEKLPwsk0ia7604Uq9vOXO8kiWuusiKTI1V9KrdY%2ffiu7xqkQ8JrT7v0Pf7aHPlmu9  
pB7pObUqN0jp%2fBA8FXxEvPKIKL%2bidkaSgW42CK9C08AnG1Kyj91bC5h%2bCF8nPNDgC8uZ6n1wCBOiKy3QsmXTZi7K  
YbMvHWvenxBvvUp%2bU%2bJQMPTA%2bd%2bpF2zpxdNI9w2G%2bF7Wp%2f1zAyL54n3IANR2s6Xo7sHZ8kLzaUNSSU  
jVleO0p0NEZQ4ahFx%2beF0I2VmyNEEdEhTo91nX4KQSYTVUnn9%2bmR%2f%2bWOOBFF1oQ2d4k8Fv%2fx7QD6ZhdSZz  
Tii3milVoDFBUk8syuLY04komsObTrvikoZGLWBkzW7WInhxiFxuYqP1S%2fSmUAq1ZSKO%2fE0q6q5mxtLmK1ggjqQC1bk4  
NM8Zmm4HtWtjRq4pADpON%2fnpa%2b073r2f%2fIRLcBkMzn6vgW9vH3NqqSEr0JHKhY43SEnI09tBQhAffaQnhbMsTri  
w%2faFcJGnHhallaJ%2bsKUgUisMRMgVXAAORWoBoYMLwu4du5vVrVBPnw8UURcTi19toqn1UribrbeCuuG6xd8giNRRdS  
elK9B9srS3wlmcYsrzxK5eqmnuhbDeHg0%2be8XMfL8F5fFNWJTOrlqRwfA8S1xCcn3SfWpjXu41d9ZGUFU74Y%2f5kJUKgt  
oU22cBxyujRQvRqi9qeW2%2bbg8INR8lqtCZAW6huO3Wk3dwqLrtZlo2S7mIUGyKpCCnAZ0ZSAvMHO%2fPgXpOBEC6msk  
Yx2CNCJfKnydKW8U%2bxVN1m%2fUa%2btkMh0b9MfeHG8uPXmmcs7XVC1WYm2ApK6%2fZ93eEEnFvPgNtFexxTYWwg  
1AArOvg%2bVzeWyyUJ9UpR91DmAAhJGp%2fih%2fHplpihTr5Plg7I3Gu3i97pmmKF%2bC4JKVCKpDZ3kp0T2SBuPewNzb  
OuBt4fySvldail4dEJfeVEekc1QjSmn4%2fKuB9mSI%3d

### Appendix 3. Error codes

Error codes of transaction are updated continuously, which can be searched on [merchant's admin portal](#). Please log into merchant's admin portal: System Development Management→Transaction status code query.

**※Special note:**

The following is translated by Google translation; the original is in Chinese only.



## Appendix 4. URL Encoding Reference

Special characters	After URLEncoding	System.Web.HttpUtility.UrlEncode() (.NET Framework)
-	%2d	-
_	%5f	_
.	%2e	.
!	%21	!
~	%7e	%7e
*	%2a	*
(	%28	(
)	%29	)
space	%20	+
@	%40	%40
#	%23	%23
\$	%24	%24
%	%25	%25
^	%5e	%5e
&	%26	%26
=	%3d	%3d
+	%2b	%2b
;	%3b	%3b
?	%3f	%3f
/	%2f	%2f
\	%5c	%5c
>	%3e	%3e
<	%3c	%3c
%	%25	%25
`	%60	%60
[	%5b	%5b
]	%5d	%5d
{	%7b	%7b
}	%7d	%7d
:	%3a	%3a
'	%27	%27
"	%22	%22
,	%2c	%2c
	%7c	%7c