

綠界金流

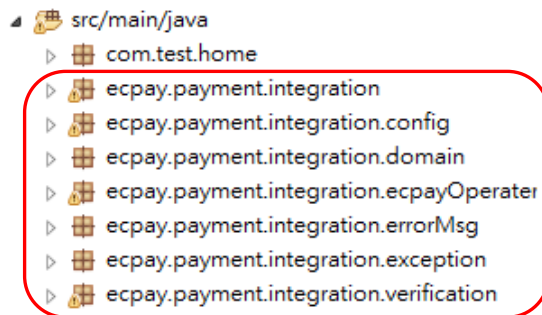
Java SDK 使用者說明

內容

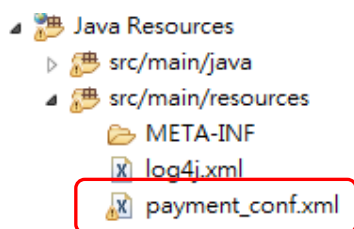
安裝 SDK.....	2
設定 SDK.....	2
使用 SDK 方法.....	3
建議的使用流程.....	4
Method 說明	5
1. <i>aioCheckOut method(Object, InvoiceObj)</i> 產生訂單 form	5
2. <i>aioCheckOutFeedback(HttpServletRequest)</i> ATM, CVS 或 BARCODE 取號結果通知.....	6
3. <i>queryTradeInfo(QueryTradeInfoObj)</i> 查詢綠界訂單	6
4. <i>queryCreditCardPeriodInfo(QueryCreditCardPeriodInfoObj)</i> 信用卡定期定額訂單查詢.....	6
5. <i>doAction method</i> 信用卡關帳/退刷/取消/放棄.....	7
6. <i>compareCheckMacValue(Hashtable<String, String>)</i> 比對 Hashtable 中的檢查碼	7
7. <i>queryTrade(QueryTradeObj)</i> 查詢信用卡單筆明細記錄.....	7
8. <i>tradeNoAio(TradeNoAioObj)</i> 下載會員對帳媒體檔.....	8
9. <i>fundingReconDetail (FundingReconDetailObj)</i> 下載信用卡撥款對帳資料檔.....	8
Sample code(範例程式碼)	9
Testing code	9
附錄.....	10

安裝 SDK

1. github 下載 SDK
2. 將 SDK 中整個 `ecpay` 資料匣放入專案中的 `src/main/java` 路徑裡面(或是 Java 的 root path 中) , 當開啟專案的時候就會自動變成 package 如圖示

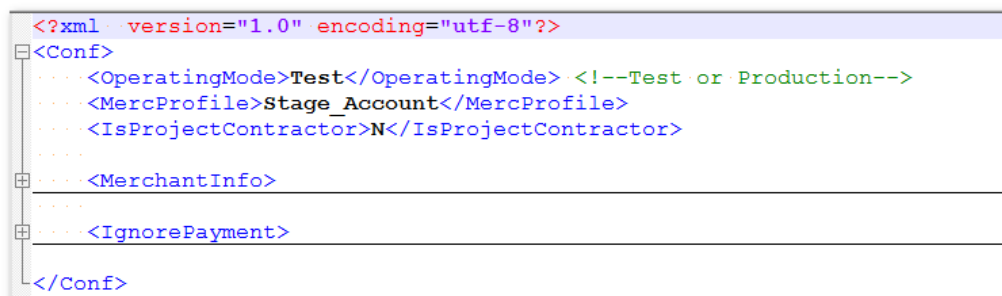


3. 將 `payment_conf.xml` 放在 `src/main/resources` 中 , 或是自定義的路徑(但要自行更改 SDK `ecpay.payment.integration` 中 `AllInOneBase.java` 建構子中預設的檔案讀取路徑)



設定 SDK

File : `payment_conf.xml`



主要的設定分為以下五節:

1. *OperatingMode*

決定 SDK 中 method 串接時連結的 API URL 位址 , 僅可設定為 Test 或 Production 。

2. MercProfile

依 OperatingMode 定義，設定 SDK 中串接要使用的 MID[會員編號]，名稱須對應到下方 MerchantInfo 中 MInfo 的 name 屬性。

3. IsProjectContractor

指定是否為專案平台商，若設定為 Y 則會將目前設定的 MID 帶入 API 的 PlatformID 參數中，設為 N 則是帶入空字串。

4. MerchantInfo

```
<MerchantInfo>
  <MInfo name="Production_Account">
    <MerchantID></MerchantID>
    <HashKey></HashKey>
    <HashIV></HashIV>
  </MInfo>
  <MInfo name="Stage_Account">
    <MerchantID>2000132</MerchantID>
    <HashKey>5294y06JbISpM5x9</HashKey>
    <HashIV>v77hoKGq4kWxNNIS</HashIV>
  </MInfo>
</MerchantInfo>
```

MInfo 可依上方 OperatingMode 分別定義一組 MID、HashKey 及 HashIV。至少要有一組 name 屬性對應到上方 MercProfile 設定值。

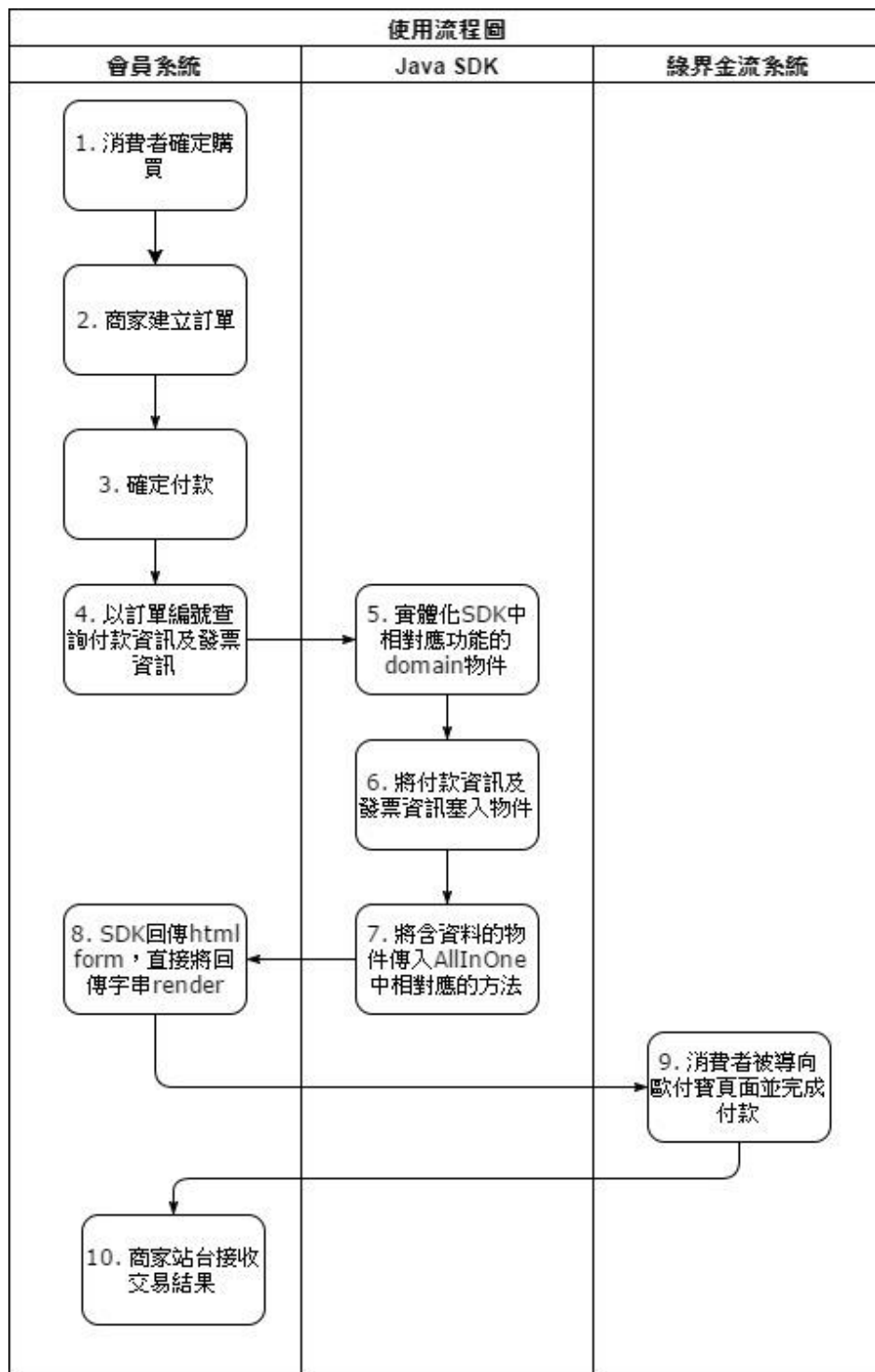
5. IgnorePayment

當使用 AioCheckOutALL 時欲排除的付款方式。SDK 會將讀取到的值帶入 IgnorePayment 參數當中。

使用 SDK 方法

1. 使用前先透過呼叫建構子 AllInOne 來實體化(new) AllInOne 類別，**建構子參數請填入 log4j.properties 的位置路徑**，但若不需特別開啟 log 或記錄 log 請將參數放入空字串(非 debug 階段請放入空字串，或是在 log4j.properties 設定成不記錄 log 到檔案中，**避免敏感資料暴露**)
2. 實體化(new) 出所需要的 domain 物件
3. 將資料透過物件的 setter method 填入 domain 物件中
4. 呼叫 AllInOne 中 API 串接所需要的 method，並將 domain 物件傳入使用並得到回傳字串

建議的使用流程



1. 消費者確定購買品項及填入發票資訊(如欲開立電子發票)後確定建立商家訂單
2. 商家站台建立訂單，產生唯一的訂單編號。
3. 消費者確定付款(訂單編號傳送至商家站台 API)
4. 商家站台以訂單編號查詢出相關的付款資訊及發票資訊
5. 實體化(new)SDK 中相對應功能的 domain 物件
6. 將付款資訊與發票資訊透過物件的 setter method 填入物件屬性當中
7. 透過物件傳遞付款資訊及發票資訊給 SDK AllInOne class 中相對應的方法
8. 商家站台 API 將 SDK 方法回傳的 html 直接 render
9. 消費者被導向至綠界付款頁面完成付款。
10. 商家站台接收交易結果，檢查交易回傳資訊是否正確。
11. 商家站台藉由 SDK 查詢用方法檢查綠界訂單資訊。

Method 說明

方法一覽

Method Name	Usage
<i>aioCheckOut method(Object, InvoiceObj)</i> 產生訂單 <i>form</i>	Java SDK 產生訂單 form
<i>aioCheckOutFeedback(HttpServletRequest)</i> ATM, CVS 或 <i>BARCODE</i> 取號結果通知	Java SDK ATM、CVS 或 <i>BARCODE</i> 取號結果通知
<i>queryTradeInfo(QueryTradeInfoObj)</i> 查詢綠界訂單	Java SDK 訂單查詢
<i>queryCreditCardPeriodInfo(QueryCreditCardPeriodInfoObj)</i> 信用卡定期定額訂單查詢	Java SDK Credit 定期定額查詢
<i>doAction method</i> 信用卡關帳/退刷/取消/放棄	Java SDK 信用卡關帳/退刷/取消/放棄
<i>compareCheckMacValue(Hashtable<String, String>)</i> 比對 <i>Hashtable</i> 中的檢查碼	Java SDK 檢查 <i>Hashtable</i> 中的檢查碼是否正確
<i>queryTrade(QueryTradeObj)</i> 查詢信用卡單筆明細記錄	Java SDK 查詢信用卡單筆明細紀錄
<i>tradeNoAio(TradeNoAioObj)</i> 下載會員對帳媒體檔	Java SDK 對帳 CSV
<i>fundingReconDetail (FundingReconDetailObj)</i> 下載信用卡撥款對帳資料檔	Java SDK 下載信用卡撥款對帳資料

1. aioCheckOut method(Object, InvoiceObj) 產生訂單 form

傳入參數說明

參數資料型別	資料
Object	為 AioCheckOut 物件，含有 AioCheckOutALL, AioCheckOutATM, AioCheckOutCVS, AioCheckOutBARCODE, AioCheckOutDevide, AioCheckOutOneTime, AioCheckOutPeriod, AioCheckOutWebATM 等不同物件，請根據相對應的功能給予相對應的物件，以免發生非預期的結果。 物件屬性請參考 https://www.ecpay.com.tw/Content/files/ecpay_011.pdf PaymentType, ChoosePayment, EncryptType 屬性請忽略，SDK 會自動帶入 若為特約合作平台商， PlatformID 請從 XML 設定，並填寫 MerchantID 屬性 若非特約合作平台商，請勿填 MerchantID ，SDK 會自動從 XML 設定的值設定
InvoiceObj	為開立發票物件，物件屬性請參考 https://www.ecpay.com.tw/Content/files/ecpay_011.pdf CustomerAddr, CustomerName, CustomerEmail, InvoiceItemName, InvoiceItemWord 等不需先做 UrlEncode ，直接帶入資料即可 此物件非必填，若不開發票則請帶入 null 即可。

以下參數為傳遞給付款方法 AioCheckOut 的基本參數。

AioCheckOutXXX 物件必帶，InvoiceObj 物件可帶可不帶。若不需開發票，則 InvoiceObj 請傳入 null 即可；若需開發票，則將發票資訊填入 InvoiceObj 中並傳入 method，method 會自動將屬性加入。

物件內的所有屬性值都為 String 格式，請勿放入 null 或其他非法型態。

此方法最後回傳一個 html form 由 JavaScript submit 的 String，只要將其放在網頁上即可(欲詳細了解如何放置在網頁可參閱 PrintWriter，或查詢該系統 framework 將字串放入網頁中 render 的相關資訊與文件)。

回傳值說明

回傳格式	資料
String	組成 html form 並使用 JavaScript submit 的字串

2. aioCheckOutFeedback(HttpServletRequest) ATM, CVS 或 BARCODE 取號結果通知

傳入參數說明

參數資料型別	資料
HttpServletRequest	接收到 Server POST 的資料物件

使用 ATM/CVS/BARCODE 付款方式建立訂單完成後，綠界會以 ServerPOST 方式傳送至訂單資料設定的回傳付款網址 PaymentInfoURL。

透過此方法，傳入 request，此方法會根據傳入的資訊將其放入相應的物件中，並檢查 CheckMacValue 是否正確，以確保資料未被篡改

回傳值說明

回傳格式	資料
Object	依使用者交易訂單的種類，物件可分為 ATMRequestObj 與 CVSOrBARCODERequestObj，請使用相應的物件接收以免出錯 物件屬性請參考 https://www.ecpay.com.tw/Content/files/ecpay_011.pdf

3. queryTradeInfo(QueryTradeInfoObj) 查詢綠界訂單

傳入參數說明

參數資料型別	資料
QueryTradeInfoObj	物件屬性請參考 https://www.ecpay.com.tw/Content/files/ecpay_011.pdf TimeStamp 屬性請忽略，SDK 會自動帶入 若為合作平台商，PlatformID 請從 XML 設定，並填寫 MerchantID 屬性 若非合作平台商，請忽略不填 MerchantID，SDK 會自動從 XML 所設的值設定

回傳值說明

回傳格式	資料
String	查尋回傳的字串

4. queryCreditCardPeriodInfo(QueryCreditCardPeriodInfoObj) 信用卡定期定額訂單查詢

傳入參數說明

參數資料型別	資料
QueryCreditCardPeriodInfoObj	物件屬性請參考 https://www.ecpay.com.tw/Content/files/ecpay_011.pdf TimeStamp, MerchantID 屬性請忽略，SDK 會自動帶入

回傳值說明

回傳格式	資料
String	查尋回傳的字串

5. doAction method 信用卡關帳/退刷/取消/放棄

傳入參數說明

參數資料型別	資料
DoActionObj	物件屬性請參考 https://www.ecpay.com.tw/Content/files/ecpay_011.pdf 若為合作平台商，PlatformID 請從 XML 設定，並填寫 MerchantID 屬性 若非合作平台商，請忽略不填 MerchantID，SDK 會自動從 XML 所設的值設定

回傳值說明

回傳格式	資料
String	API 回傳的字串

6. *compareCheckMacValue(Hashtable<String, String>)* 比對 Hashtable 中的檢查碼

傳入參數說明

參數資料型別	資料
Hashtable<String, String>	資料所組成的字典(含 CheckMacValue)

此方法用來檢查是否資料被竄改。Hashtable 中一定要有 CheckMacValue，否則會出錯

回傳值說明

回傳格式	資料
boolean	判斷回傳檢查碼與 SDK 所產生的檢查碼是否一致

7. *queryTrade(QueryTradeObj)* 查詢信用卡單筆明細記錄

傳入參數說明

參數資料型別	資料
QueryTradeObj	物件屬性請參考 https://www.ecpay.com.tw/Content/files/ecpay_011.pdf MerchantID 屬性請忽略，SDK 會自動帶入

回傳值說明

回傳格式	資料
String	API 回傳的字串

8. *tradeNoAio(TradeNoAioObj)* 下載會員對帳媒體檔

傳入參數說明

參數資料型別	資料
TradeNoAioObj	物件屬性請參考 https://www.ecpay.com.tw/Content/files/ecpay_011.pdf MerchantID 屬性請忽略，SDK 會自動帶入

回傳值說明

回傳格式	資料
String	API 回傳的字串

9. *fundingReconDetail* (*FundingReconDetailObj*) 下載信用卡撥款對帳資料檔

傳入參數說明

參數資料型別	資料
FundingReconDetailObj	物件屬性請參考 https://www.ecpay.com.tw/Content/files/ecpay_011.pdf MerchantID 屬性請忽略・SDK 會自動帶入

回傳值說明

回傳格式	資料
String	API 回傳的字串

Sample code(範例程式碼)

注意事項:

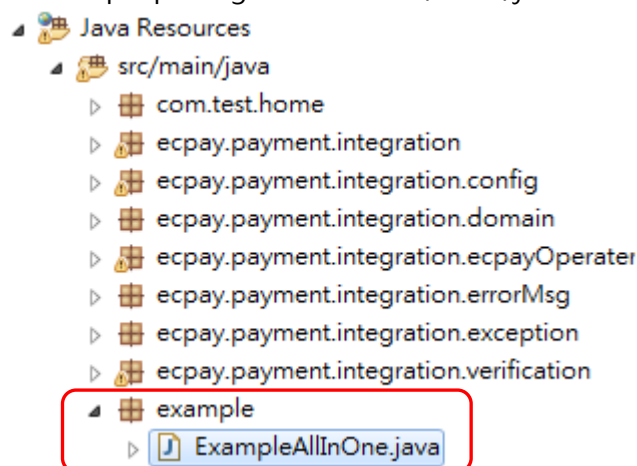
1. 範例程式碼為提供串接用戶參考用之範例程式碼，請勿直接套用在您的商業環境。
2. 考慮到一般電商運用以及基本的安全性，範例程式碼假設了串接者已經具備某些外部函式支援資料的取得以及標準化。因此直接將範例碼複製到您的環境中可能是無法正常產出結果的。

Testing code

提供使用者在 Java web project 中測試 API 串接是否正常的程式碼

使用方法

1. 將 example package 放入專案 src/main/java 中



2. ExampleAllInOne.java 修改想要填入的值(規則可參考本文件)
3. 將 ecpay.payment.integration package 中的 AllInOneBase.java 建構子中的 ClassLoader 三行並 try, catch 換成 paymentConfPath 兩行(測試完記得改回來以免出錯!)

```
33 public AllInOneBase(){
34     // try{
35         Document doc;
36         /* when using web project*/
37         // ClassLoader classLoader = Thread.currentThread().getContextClassLoader();
38         // String configPath = URLDecoder.decode(classLoader.getResource("/payment_conf.xml").getPath(), "UTF-8");
39         // doc = EcpayFunction.xmlParser(configPath);
40         /* when using testing code*/
41         String paymentConfPath = "./src/main/resources/payment_conf.xml";
42         doc = EcpayFunction.xmlParser(paymentConfPath);
43
44         doc.getDocumentElement().normalize();
45     }
```

4. 右鍵點選 ExampleAllInOne.java -> Run As -> Java Application
5. 檢查回傳的字串(應該要回傳一些 Query 的結果和建立訂單使用的 form)
6. 將建立訂單的 form 貼在文字編輯器(如記事本)中並儲存(副檔名請改為 html)
7. 雙擊該檔案應可打開看到綠界結帳頁面(不用在意錯誤訊息，可以看到綠界頁面即可)

注意事項

1. 測試程式碼僅用於開發階段時驗證 API 串接用，請勿直接套用在您的商業環境。

2. 各 API 串接 method 中可能有某些參數是需要由 WebApp 內部直接提供，以避免舞弊，**請務必避免將測試程式碼直接套用後，由客戶端瀏覽器將所有參數 POST 上來。**

保護措施

請注意設定檔讀取權限

Troubleshooting

附錄

欲詳細了解規格請參考 https://www.ecpay.com.tw/Content/files/ecpay_011.pdf