

1. 下載綠界 SDK 程式(https://www.ecpay.com.tw/Service/API_Dwnld)

金流應用

金流整合串接

提供Web、SDK及API介接規格，申請金流程式串接取得加密金鑰，步驟簡易輕鬆串接。

SDK 程式

付款方式包含信用卡(一次付清、分期付款、定期定額)、Web ATM、ATM 櫃員機、超商代碼、超商條碼(Java除外)、Apple Pay等介接規格。

PHP Java Ruby Node.js C#(.Net) Python

API 介接文件

提供完整的交易介接技術及多種收款方式供合作特店選擇，包含信用卡、Web ATM、ATM 櫃員機、超商代碼、超商條碼等；主要說明各種收款方式介接規格說明，與綠界科技金流服務提供各種交易 API 的使用時機及方法。

完整版-全方位金流API技術文件 All-In-One API document(EN)

36 commits1 branch0 packages2 releases3 contributors

Branch: masterNew pull requestFind fileClone or download

adelachang0418 修改版本號

Doc	更新JAVA SDK 金流文件	
ecpay/payment/integration	修改版本號	
example	調整付款方式範例	
jars	綠界 全方位(All In One)金流介接Java版	3 years ago
README.md	Update README.md	4 months ago
payment_conf.xml	調整付款方式範例	9 months ago

Clone with HTTPS

Use Git or checkout with SVN using the web URL.

https://github.com/ECpay/ECpayAIO_Java.git

Open in Desktop Download ZIP

2. 將下載完的檔案解壓縮，放進專案內

將 ecpay 資料夾整個放入專案中 src/main/java 路徑下

ECpayAIO_Java-master

名稱	修改日期	類型	大小
Doc	2020/03/04 19:15	檔案資料夾	
ecpay	2020/03/04 19:15	檔案資料夾	
example	2020/03/04 19:15	檔案資料夾	
jars	2020/03/04 19:15	檔案資料夾	
payment_conf	2020/03/04 19:15	XML Document	1 KB
README.md	2020/03/04 19:15	MD 檔案	2 KB

speakitup [speakitup master]

JAX-WS Web Services

Java Resources

src/main/java

com.web.speakitup

ecpay.payment.integration

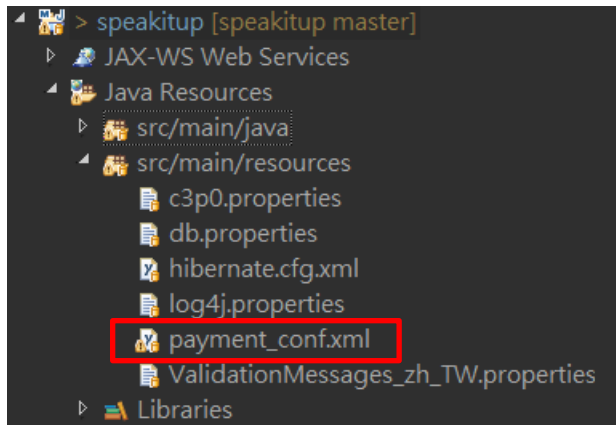
src/main/resources

Libraries

JavaScript Resources

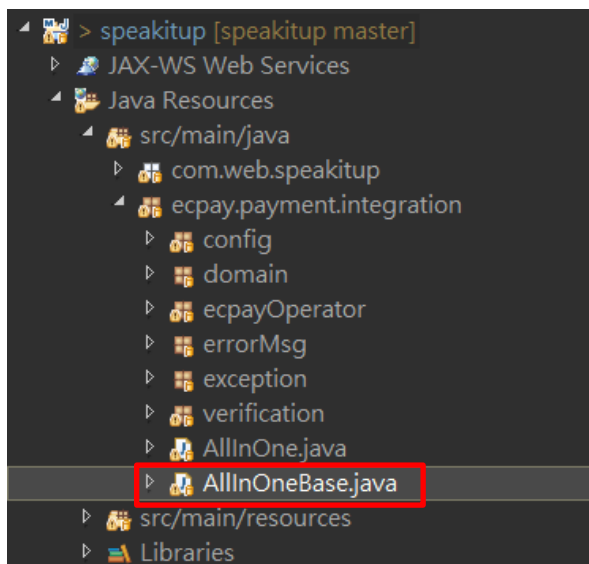
將 payment_conf.xml 檔放入專案中 src/main/resources 路徑下

> ECPayAIO_Java-master				
	名稱	修改日期	類型	大小
Files	Doc	2020/03/04 19:15	檔案資料夾	
	ecpay	2020/03/04 19:15	檔案資料夾	
	example	2020/03/04 19:15	檔案資料夾	
	jars	2020/03/04 19:15	檔案資料夾	
	payment_conf	2020/03/04 19:15	XML Document	1 KB
	README.md	2020/03/04 19:15	MD 檔案	2 KB



3. 改寫 xml 檔路徑

打開專案中 ecpay 內的 AllInOneBase.java



修改 paymentConfPath 為 payment_conf.xml 檔的絕對路徑

(專案資料夾路徑/src/main/resources/payment_conf.xml)

```

public AllInOneBase(){
    try{
        Document doc;
        /* when using web project*/
        ClassLoader classLoader = Thread.currentThread().getContextClassLoader();
        String configPath = URLDecoder.decode(classLoader.getResource("/payment_conf.xml").getPath(), "UTF-8");
        doc = EcpayFunction.xmlParser(configPath);
        /* when using testing code*/
        String paymentConfPath = "./src/main/resources/payment_conf.xml";
        String paymentConfPath = "C:/TOOLS/JAVA/workspace/speakitup/src/main/resources/payment_conf.xml";
        doc = EcpayFunction.xmlParser(paymentConfPath);
    }
}

```

4. 修改 payment_conf.xml 檔

(此處為忽略信用卡以外付款方式)

```

<?xml version="1.0" encoding="utf-8"?>
<Conf>
    <OperatingMode>Test</OperatingMode> <!--Test or Production-->
    <MercProfile>Stage_Account</MercProfile>
    <IsProjectContractor>N</IsProjectContractor>

    <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>

    <IgnorePayment>
        <!-- <Method>Credit</Method> -->
        <Method>WebATM</Method>
        <Method>ATM</Method>
        <Method>CVS</Method>
        <Method>BARCODE</Method>
    </IgnorePayment>

</Conf>

```

5. 設定綠界參數 (綠界介接完整文件：https://www.ecpay.com.tw/Content/files/ecpay_011.pdf)

取得測試資料 (p.17)

欄位說明	欄位內容		
特店編號(MerchantID)	2000132	2000214(非 OTP)	
平台商編號 (PlatformID)			3002599
廠商管理後台登入帳號/ 密碼	stagetest1234/test1234	Stage2000214/test1234	stagetest2/test1234
身分證件末四碼/統一編號	53538851		3609
廠商管理後台 測試環境	https://vendor-stage.ecpay.com.tw 此網站可提供：1. 查詢 ecpay 訂單，2. 模擬付款並通知特店 若執行模擬付款功能，無法收到付款通知時，請留意此表後 注意事項 。		
ALL IN ONE 介接 HashKey	5294y06JbISpM5x9		spPjZn66i0OhqJsQ
ALL IN ONE 介接 HashIV	v77hoKGq4kWxNNIS		hT5OJckN45isQTTs
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		
信用卡測試有效月/年	輸入的 MM/YYYY 值請大於現在當下時間的月年，例如在 2016/04/20 當天作測試，請設定 05/2016(含)之後的有效月年，否則回應刷卡失敗。		
對外 IP	請參考 綠界 FAQ：是否可限制只允許綠界 IP 連入公司的 Server		

根據付款方式選擇類別

(此處 AioCheckOutDevide 為使用信用卡)

```

/* 準備前往綠界 */
@GetMapping("/goECPay")
public void goECPay(Model model, HttpServletRequest request, HttpServletResponse response, HttpSession session)
    throws IOException {
    OrderBean ob = (OrderBean) request.getAttribute("orderBean");
    // 設定金流
    AllInOne aio = new AllInOne("");
    AioCheckOutDevide aioCheck = new AioCheckOutDevide();
    /* 特店編號 */
    aioCheck.setMerchantID("2000214");
    /* 特店交易時間 */
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");
    sdf.setLenient(false);
    aioCheck.setMerchantTradeDate(sdf.format(new Date()));
    /* 交易金額 */
    aioCheck.setTotalAmount(ob.getTotalPrice().toString());
    /* 交易描述 */
    aioCheck.setTradeDesc("speakitup");
    /* 商品名稱 */
    aioCheck.setItemName("500");
    /* 特店交易編號 */
    aioCheck.setMerchantTradeNo("testSpeakitup" + ob.getOrderNo());
    /* 付款完成通知回傳網址 */
    aioCheck.setReturnURL("https://speakitup.nctu.me/order/returnURL");
    /* Client端回傳付款結果網址 */
    aioCheck.setOrderResultURL("https://speakitup.nctu.me/order/showHistoryOrder");
    // 輸出畫面
    PrintWriter out = response.getWriter();
    response.setContentType("text/html");
    out.print(aio.aioCheckOut(aioCheck, null));
}

```

設定必填資料 (p.19)

特店編號(MerchantID)輸入測試資料，其他按照文件內格式設定

！特店交易編號(MerchantTradeNo)不能重複！

```
/* 準備前往綠界 */
@GetMapping("/goECPay")
public void goECPay(Model model, HttpServletRequest request, HttpServletResponse response, HttpSession session)
    throws IOException {
    OrderBean ob = (OrderBean) request.getAttribute("orderBean");
    // 設定金流
    AllInOne aio = new AllInOne("");
    AioCheckOutDevide aioCheck = new AioCheckOutDevide();
    /* 特店編號 */
    aioCheck.setMerchantID("2000214");
    /* 特店交易時間 */
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");
    sdf.setLenient(false);
    aioCheck.setMerchantTradeDate(sdf.format(new Date()));
    /* 交易金額 */
    aioCheck.setTotalAmount(ob.getTotalPrice().toString());
    /* 交易描述 */
    aioCheck.setTradeDesc("speakitup");
    /* 商品名稱 */
    aioCheck.setItemName("500");
    /* 特店交易編號 */
    aioCheck.setMerchantTradeNo("testSpeakitup" + ob.getOrderNo());
    /* 付款完成通知回傳網址 */
    aioCheck.setReturnURL("https://speakitup.nctu.me/order/returnURL");
    /* Client端回傳付款結果網址 */
    aioCheck.setOrderResultURL("https://speakitup.nctu.me/order/showHistoryOrder");
    // 輸出畫面
    PrintWriter out = response.getWriter();
    response.setContentType("text/html");
    out.print(aio.aioCheckOut(aioCheck, null));
}
```

輸出給綠界(如需開立發票，此處 null 需另外設定 InvoiceObj 物件 p.28)

```
/* 準備前往綠界 */
@GetMapping("/goECPay")
public void goECPay(Model model, HttpServletRequest request, HttpServletResponse response, HttpSession session)
    throws IOException {
    OrderBean ob = (OrderBean) request.getAttribute("orderBean");
    // 設定金流
    AllInOne aio = new AllInOne("");
    AioCheckOutDevide aioCheck = new AioCheckOutDevide();
    /* 特店編號 */
    aioCheck.setMerchantID("2000214");
    /* 特店交易時間 */
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");
    sdf.setLenient(false);
    aioCheck.setMerchantTradeDate(sdf.format(new Date()));
    /* 交易金額 */
    aioCheck.setTotalAmount(ob.getTotalPrice().toString());
    /* 交易描述 */
    aioCheck.setTradeDesc("speakitup");
    /* 商品名稱 */
    aioCheck.setItemName("500");
    /* 特店交易編號 */
    aioCheck.setMerchantTradeNo("testSpeakitup" + ob.getOrderNo());
    /* 付款完成通知回傳網址 */
    aioCheck.setReturnURL("https://speakitup.nctu.me/order/returnURL");
    /* Client端回傳付款結果網址 */
    aioCheck.setOrderResultURL("https://speakitup.nctu.me/order/showHistoryOrder");
    // 輸出畫面
    PrintWriter out = response.getWriter();
    response.setContentType("text/html");
    out.print(aio.aioCheckOut(aioCheck, null));
}
```

6. 用 post 承接綠界回傳值(RtnCode==1 為交易成功 p.36)

```
// 綠界回傳資料
@PostMapping("/returnURL")
public void returnUrl(@RequestParam("MerchantTradeNo") String MerchantTradeNo,
    @RequestParam("RtnCode") int RtnCode, @RequestParam("TradeAmt") int TradeAmt, HttpServletRequest request) {
    // 交易成功
    if ((request.getRemoteAddr().equalsIgnoreCase("175.99.72.1")
        || request.getRemoteAddr().equalsIgnoreCase("175.99.72.11")
        || request.getRemoteAddr().equalsIgnoreCase("175.99.72.24")
        || request.getRemoteAddr().equalsIgnoreCase("175.99.72.28")
        || request.getRemoteAddr().equalsIgnoreCase("175.99.72.32")) && RtnCode == 1) {
        String orderIdStr = MerchantTradeNo.substring(13);
        int orderId = Integer.parseInt(orderIdStr);
        OrderBean ob = orderService.getOrder(orderId);
        ob.setStatus("待出貨");
    }
}
```

7. 用 post 承接交易完成後的頁面跳轉(此處付款完畢後頁面跳轉至歷史訂單)

```
/* 查詢歷史訂單 */
@PostMapping("/showHistoryOrder")
public String showECPAYHistoryOrder(Model model, HttpSession session) {
    // 取得使用者資料(MemberBean)
    MemberBean mb = (MemberBean) session.getAttribute("LoginOK");
    Integer memberId = mb.getId();
    // 取得使用者的訂單資料(OrderBean)
    List<OrderBean> orders = orderService.getMemberOrders(memberId);

    // 取出訂單詳細資料(OrderItemBean)
    Map<Integer, Set<OrderItemBean>> orderItemGroup = new HashMap<Integer, Set<OrderItemBean>>();
    for (OrderBean ob : orders) {
        int orderNo = ob.getOrderNo();
        Set<OrderItemBean> OrderItemBeans = ob.getOrderItems();
        orderItemGroup.put(orderNo, OrderItemBeans);
    }

    model.addAttribute("order_list", orders);
    model.addAttribute("orderItem_map", orderItemGroup);

    return "order/historyOrder";
}
```