
LINE Pay 整合指南

外部商家用

著作權

Copyright © LINE Corp. All Rights Reserved.

本文件為 LINE Corp. 之智慧財產，嚴禁違法將本文的部分或全部進行未經授權的拷貝或發放。

本文之目的僅為提供資訊。LINE Corp. 雖期待本文所記載資訊的完整性和正確性，但無法對發生問題的內容之錯誤，以及疏漏部分負責。因此，使用本文時所產生的責任，皆須由用戶承擔。LINE Corp. 不做任何明示及暗示保證。包含相關網址在內，本文所言及之軟體商品或商品，皆遵從其所有者所屬地區的著作權法，未遵守相關法律而導致的所有後果及責任，須由用戶全面承擔。

LINE Corp. 將可能不經預告逕行變更本文之內容。

文件資訊

關於本文件

本文件將說明如何開發 LINE Pay 商家提供 LINE Pay 付款的流程。

對象

使用本文的對象必須為 LINE Pay 商家伺服器/網站開發者，且理解 HTTP 通訊及 RESTful API。

聯絡資訊

如果您對於本文件內容有任何疑問，或發現任何錯誤，請透過以下方式聯絡我們：

LINE Pay 技術支援 (dl_lp_tech_support@linecorp.com)

修訂記錄

版本	日期	變更記錄
1.0.0	2014.11.10	初稿
1.0.1	2014.12.01	API之修正、添加附錄
1.0.2	2015.01.15	更新授權、請款API
1.0.3	2015.01.20	add error code
1.0.4	2015.03.01	OTK功能、以及 ConfirmUrl之 Browser設定功能 使用Sandbox時候的畫面例子
1.0.4-1	2015.05.26	Currency修正、Product Image參數修正
1.0.5	2015.06.16	付款流程更新 多加 oneTimeKey 說明 多加 API 變更紀錄 多加 [附錄] 之 LINE Pay server IP 資訊以及二維碼、條碼測試方法
1.0.6	2015.07.30	全新定義可嘗試重複呼叫之 returnCode

文件慣例

原始碼之標記方式

原始碼文字將以灰底黑字的方式呈現。

```
COPYDATASTRUCT st;  
st.dwData = PURPLE_OUTBOUND_ENDING;  
st.cbData = sizeof(pp);  
st.lpData = &pp;  
::SendMessage(GetTargetHwnd(), WM_COPYDATA, (LPARAM)this->m_hWnd, (LPARAM)&st);
```

目錄

簡介	9
關於 LINE Pay	9
LINE Pay 商家註冊流程	10
LINE Pay 功能	11
一般付款	11
自動付款	11
一般功能	11
LINE Pay 付款流程	12
LINE Pay 整合流程	13
一般付款	13
自動付款	19
如何使用 LINE Pay 的各項 API	20
API 變更紀錄	21
2015-06-24	21
2015-04-30	21
付款reserve	22
付款reserve API	22
付款reserve API 規格	22
使用付款reserve API 的範例	25
付款confirm	27
付款confirm API	27
付款confirm API 規格	27
使用付款 confirm API 的範例	29
請款	31
請款 API	31
請款 API 規格	31
使用請款 API 的範例	33
授權作廢	34
授權作廢 API	34
授權作廢 API 規格	34

使用授權作廢 API 的範例	35
查看授權記錄	36
取得查看授權記錄 API	36
取得查看授權記錄 API 規格	36
使用取得查看授權記錄 API 的範例	37
查看付款紀錄	39
取得查看付款紀錄 API	39
取得查看付款紀錄 API 規格	39
使用取得查看付款紀錄 API 的範例	41
已付款項目的退款	43
退款 API	43
退款 API 規格	43
使用退款 API 的範例	44
查看 regKey 狀態	45
查看 regKey 狀態 API	45
查看 regKey 狀態 API 規格	45
使用查看 regKey 狀態 API 的範例	46
自動付款	47
自動付款 API	47
自動付款 API 規格	47
使用自動付款 API 的範例	49
註銷 regKey	50
註銷 regKey API	50
註銷 regKey API 規格	50
使用註銷 regKey API 的範例	50
附錄	52
LINE Pay 錯誤代碼	52
paymentUrl 應用程式使用指南	56
對於 Android 應用程式	56
對於 iOS 應用程式	57
confirmUrl 詳細說明以及例外狀況	58
confirmUrl 詳細說明	58
confirmUrl 例外狀況	58
伺服器對伺服器呼叫 confirmUrl 時	59
confirmUrl 規格	59
使用 confirmUrl 的範例	59
Sandbox 使用方法	60
Sandbox PC 付款	60
Sandbox 手機付款	61
Sandbox OneTimeKey 付款	61

表單與圖表

表單一覽

表 1 付款reserve API 規格	22
表 2 付款reserve API 請求參數	22
表 3 付款reserve API 回應	25
表 4 付款confirm API 規格	27
表 5 付款confirm API 請求參數	27
表 6 付款confirm API 回應	28
表 7 請款 API 規格	31
表 8 請款 API 請求參數	31
表 9 請款 API 回應	32
表 10 授權作廢 API 規格	34
表 11 授權作廢 API 回應	34
表 12 檢視查看授權記錄API規格	36
表 13 取得查看授權記錄 API 參數	36
表 14 取得查看授權記錄 API 回應	37
表 15 檢視查看付款紀錄 API 規格	39
表 16 取得查看付款紀錄 API 參數	40
表 17 取得查看付款紀錄 API 回應	40
表 18 退款 API 規格	43
表 19 退款 API 參數	43
表 20 退款 API 回應	44
表 21 查看 regKey 狀態 API 規格	45
表 22 查看是否可使用 regKey 之 API Query 參數	46
表 23 查看 regKey 狀態回應	46
表 24 自動付款 API 規格	47
表 25 自動付款 API 參數	47
表 26 自動付款 API 回應	48
表 27 regKey 註銷 API 規格	50
表 28 註銷 regKey API 回應	50
表 29 LINE Pay 錯誤代碼	52

表 30 每API別 ReturnCode總表	54
表 31 confirmUrl API 規格	59
表 32 confirmUrl 參數	59

圖表一覽

圖 1 LINE Pay 付款流程	9
圖 2 商家註冊流程	10
圖 3 電腦的付款reserve 流程 - 付款畫面	13
圖 4 手機的付款reserve 流程 - 付款畫面	14
圖 5 電腦的付款畫面 - 付款完成	15
圖 6 手機的付款畫面 - 付款完成	16
圖 7 伺服器對伺服器呼叫 confirmUrl	17
圖 8 oneTimeKey之生成以及 reserve	18
圖 9 自動付款	19
圖 10 Sandbox PC 進行付款畫面之例	60
圖 11 Sandbox PC 進行付款畫面之例	61

簡介

本章將說明使用 LINE Pay 的基本概念。

關於 LINE Pay

LINE Pay 是一款可供 LINE 用戶在 LINE Pay 商家網站使用的付款系統。

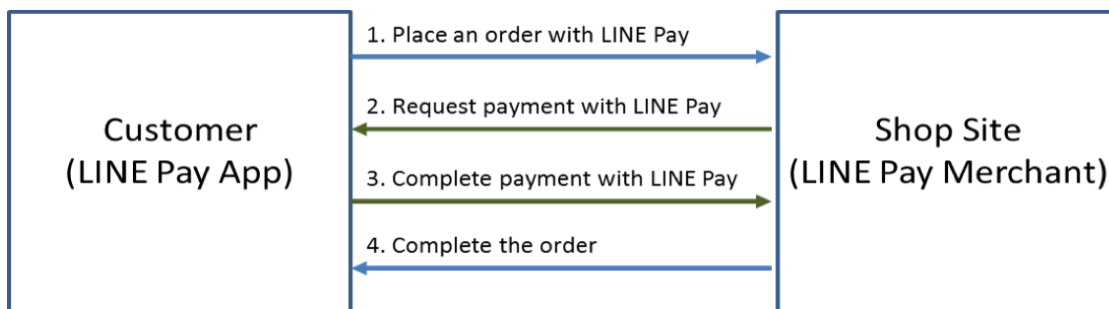


圖 1 LINE Pay 付款流程

當商家註冊成為 LINE Pay 商家後，將可以吸引全世界的 LINE 用戶成為商家的顧客。此外，商家可以透過 LINE 擴展行銷通路，以期提高業績。

想提供 LINE Pay 用戶進行付款的網站必須先註冊為 LINE Pay 商家，才能夠提供用戶使用 LINE Pay 付款。

LINE Pay 商家註冊流程

一旦商家註冊成為 LINE Pay 商家，將會收到整合金鑰與測試金鑰。商家註冊流程如下圖所示。

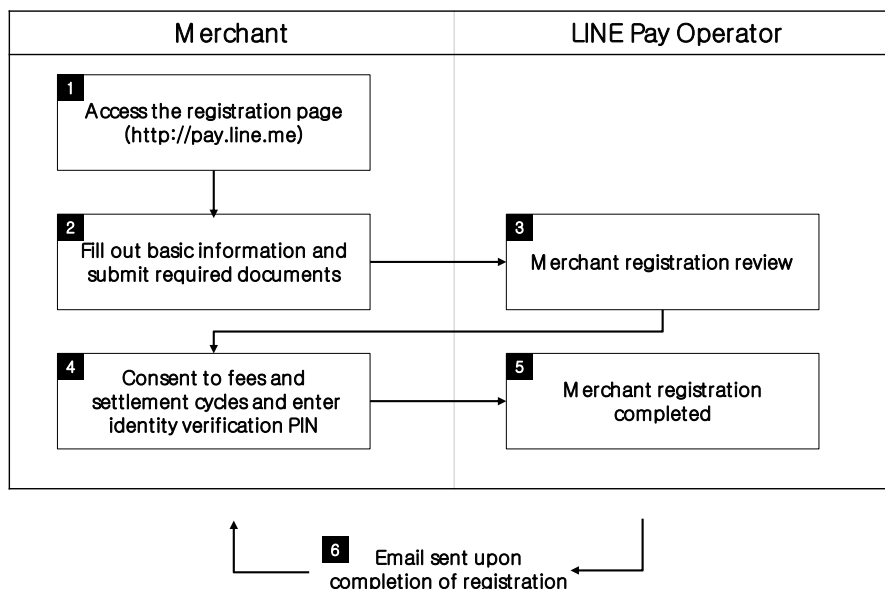


圖 2 商家註冊流程

1. 前往註冊網頁 (<http://pay.line.me>).
2. 填寫基本資料並送出必要的書面文件
3. 審核註冊商家
4. 同意手續費及結算週期，然後輸入身分認證碼
5. 完成商家註冊
6. 傳送完成註冊電子郵件

LINE Pay 功能

以下為所提供的各項功能。

一般付款

一般付款的付款方式是由 LINE Pay 用戶存取 LINE Pay 服務，選擇一種付款方式，然後輸入付款密碼來進行付款。實際商家能夠拿到收款是在商家對此筆交易進行請款後才能向顧客請款。商家可以選擇**授權與付款同時發生**，就是該筆交易發生同時完成交易。如果商家像鑑賞期之考慮，想要將付款授權與請款兩者分別處理，則應該呼叫「請款 API」。視確認密碼時所選擇的付款方式而定，金額將從買家的餘額或信用卡額度中扣除。

1. 商家伺服器呼叫「付款reserve API」。
2. 買家在 LINE Pay 付款畫面上選擇付款方式並輸入付款密碼。
3. 商家伺服器呼叫「付款confirm API」。
4. 付款交易完成。

自動付款

自動付款是在商家伺服器與 LINE Pay 伺服器之間進行，LINE Pay 用戶並未介入。當 LINE Pay 用戶第一次付款時，LINE Pay 會將自動付款的 regKey 核發給商家，然後就像一般付款一樣，經過付款保留與確認流程。此後，商家就可以使用核發的 regKey，藉由呼叫自動付款 API 來完成以下交易，不需要透過 LINE 應用程式。整個流程如下：

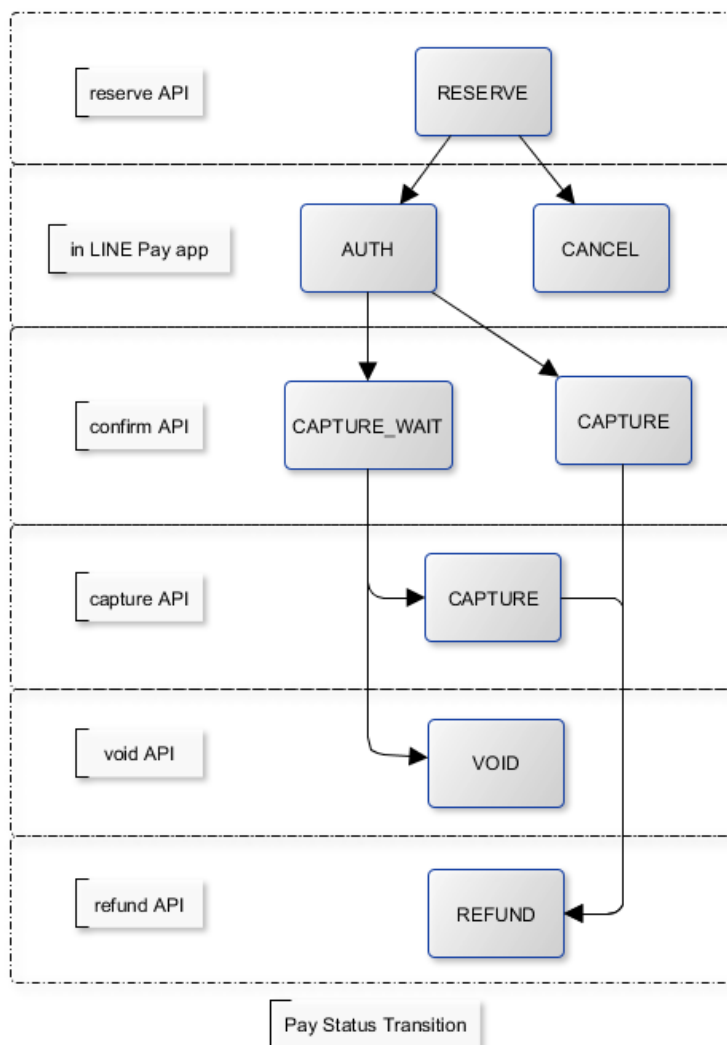
1. 第一次付款與核發 regKey (用於自動付款)
 - a. 商家伺服器呼叫付款reserve API ("payType" : "PREAPPROVED")
 - b. LINE Pay 用戶在 LINE Pay 付款畫面選擇付款方式並輸入付款密碼後，被重新導向到 confirmUrl (付款reserve API 的參數之一)。
 - c. 商家伺服器呼叫付款confirm API，並儲存傳回的 regKey。
 - d. 付款交易完成。
2. 自動付款
 - a. 商家伺服器使用 regKey 呼叫自動付款 API。
 - b. 付款交易完成。

一般功能

1. 退款
2. 取得交易詳細說明

LINE Pay 付款流程

在此章節將解釋 LINE Pay 從付款申請到完成付款的流程。



1. RESERVE：指呼叫「付款 Reserve API」時候的初始狀態
 2. AUTH：指買家在 LINE Pay app 過了選擇付款方式與密碼驗證的步驟
 3. CANCEL：指在 LINE Pay app 取消付款
 4. CAPTURE_WAIT：指等待商家在後台操作請款(只有 reserve API的“capture”參數為“false”時候才適用)
 5. CAPTURE：指請款
 6. VOID：指在「CAPTURE_WAIT」狀態下做取消該筆交易。
 7. REFUND：指請款後做取消該筆交易
- FAILED：指在 LINE Pay API處理中途遭遇失敗的狀態，在呼叫 confirm, capture, void API 時會發生

LINE Pay 整合流程

一般付款

請求付款 - 付款畫面

付款reserve 流程會因 LINE Pay 用戶的使用環境而異，分述如下：

1. 透過電腦

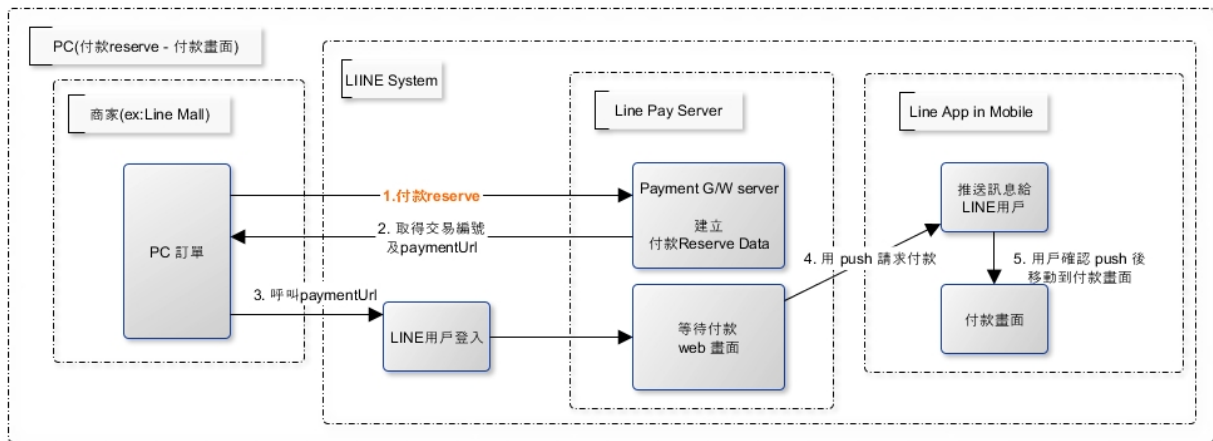


圖 3 電腦的付款reserve 流程 - 付款畫面

- LINE Pay 用戶從商家的訂購單中選擇一種付款方式。
- 商家呼叫 LINE Pay 的「付款reserve API」來建立付款reserve 資訊和取得交易編號及 "paymentUrl", 其中的交易編號為付款confirm 時的必要資訊。
- 商家呼叫在步驟 b 中收到的 paymentUrl (由 LINE Pay 提供)。
- LINE Pay 用戶經由通路網頁登入¹確認是否為 LINE 用戶。
- 當用戶成功登入 LINE 後，LINE Pay 伺服器推送付款請求到 LINE。
- LINE 用戶在 LINE 應用程式中收到付款請求後，會移動至付款畫面。

¹ 這個流程是在使用 LINE 服務前，確認用戶是否為 LINE 用戶。如果商家可以取得 LINE 用戶資訊，則略過通路網頁登入程序，並在付款reserve 時轉送 mid。

2. 透過手機

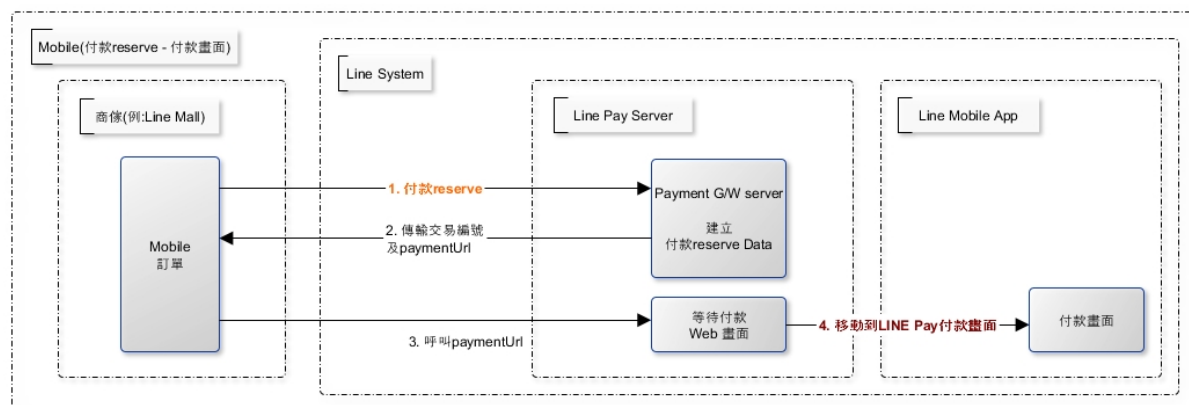


圖 4 手機的付款reserve 流程 - 付款畫面

- a. LINE Pay 用戶從商家的訂購單中選擇一種付款方式。
- b. 商家呼叫 LINE Pay 的「付款reserve API」來建立付款reserve 資訊和取得交易編號及 "paymentUrl", 其中的交易編號為付款confirm 時的必要資訊。
- c. 商家呼叫在步驟 b 中收到的 paymentUrl (由 LINE Pay 提供)。
- d. 商家使用 paymentUrl 將 LINE Pay 用戶重新導向到 LINE 應用程式中的付款畫面。視整合環境而定, paymentUrl 分成網頁與應用程式。
 - 網頁：由 LINE Pay 提供的網頁畫面。商家先檢查用戶是否已安裝 LINE 應用程式, 然後將用戶重新導向到 LINE Pay 付款畫面。
 - 應用程式：一個應用程式配置 URL, 可直接移動到 LINE Pay 付款畫面。有關如何檢查是否已安裝 LINE 應用程式, 以及可用的 LINE Pay 版本 (限 Android 平台), 請參考「paymentUrl 應用程式使用指南」。

付款畫面 - 付款完成

LINE用戶選擇LINE Pay為付款方式 (信用卡或餘額) 進行付款後, LINE Pay 用戶在完成付款授權會移動到商家在Reserve階段收到的“confirmUrl”(相應的 transactionId 為新增 Parameter, 是為了確認該筆交易)。接著, 用戶到confirmURL之後再來商家呼叫“付款confirm API”, 完成付款。

1. 透過電腦

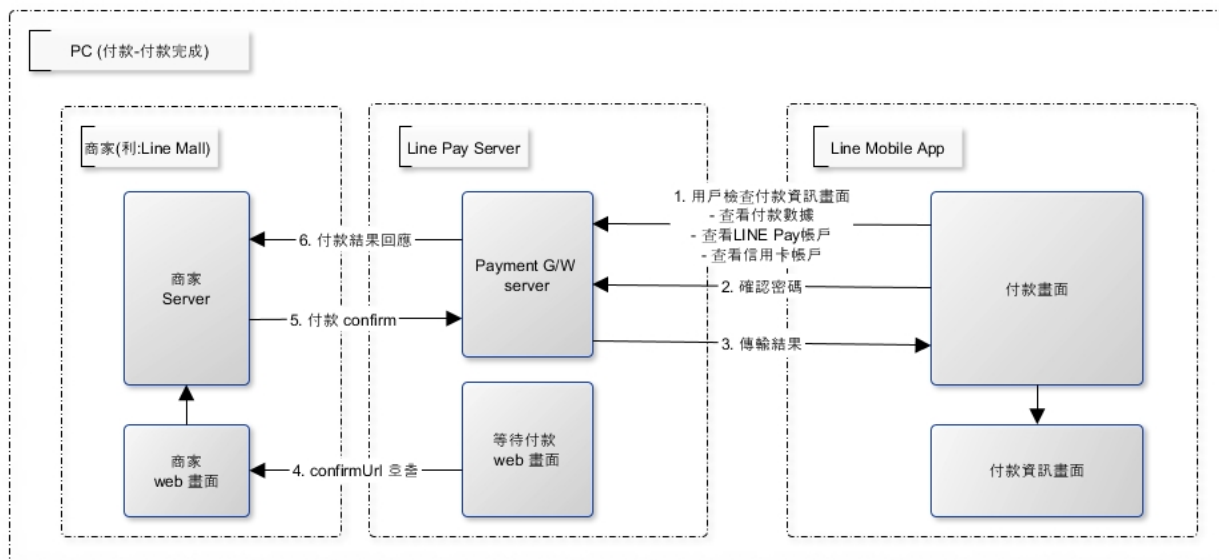


圖 5 電腦的付款畫面 – 付款完成

- LINE Pay 用戶在 LINE Pay 付款畫面上選擇付款方式並輸入密碼。
- LINE Pay 儲存付款方式資訊, 並將付款狀態變更為已授權。
- LINE Pay 用戶檢查付款資訊畫面。
- 當等待付款畫面中的交易變成可付款時, 用戶會在付款已預約時被重新導向到從商家處收到的 "confirmUrl"。
- 商家呼叫付款confirm API, 完成付款。

2. 透過手機

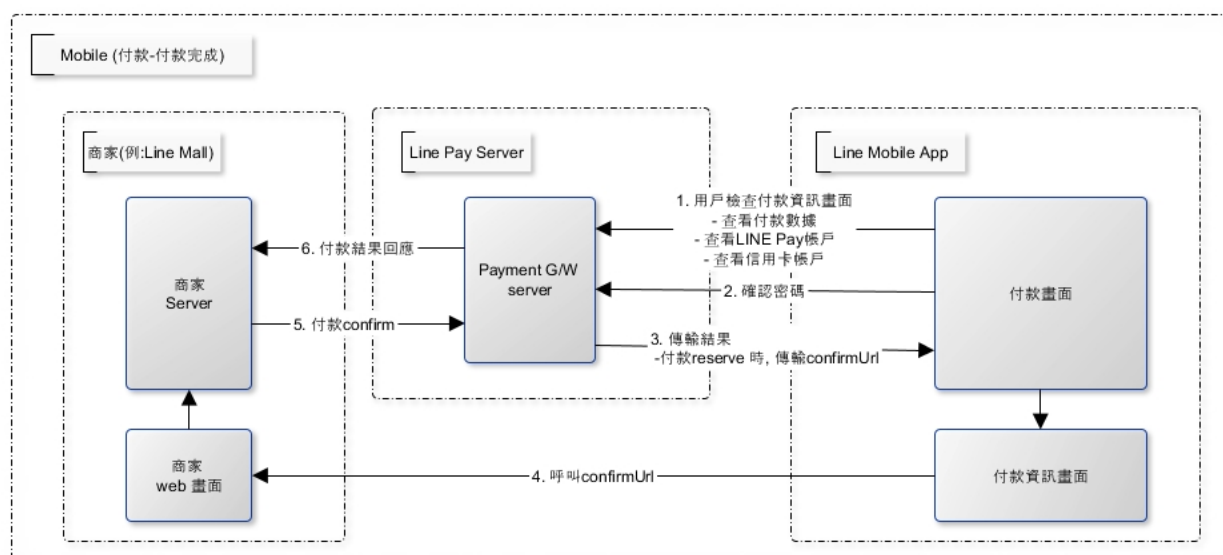


圖 6 手機的付款畫面 - 付款完成

- LINE Pay 用戶在 LINE Pay 付款畫面上選擇付款方式並輸入密碼。
- LINE Pay 儲存付款方式資訊，並將付款狀態變更為已授權。
- 檢查 LINE 應用程式中的付款資訊畫面後，LINE Pay 用戶在畫面底部按「確定」，在付款已預約時移動到從商家處收到的 "confirmUrl"。
- 商家呼叫付款confirm API，完成付款。

■ 當付款授權與請款兩者分別處理時

- 整個流程與上述類似，不過商家在呼叫付款reserve API 時，應該將 "capture" 設定為 "false"。
- 當商家呼叫付款confirm API，付款狀態會儲存為 "AUTHORIZATION"。
- 請款時，
 - 請款：呼叫請款 API 完成付款。
 - 不請款：呼叫授權作廢 API 以取消付款的授權。

伺服器對伺服器呼叫 confirmUrl 時

只能藉由商家伺服器與 LINE Pay 伺服器之間的通訊來進行付款，LINE Pay 用戶未進入商家首頁。

1. 整合流程

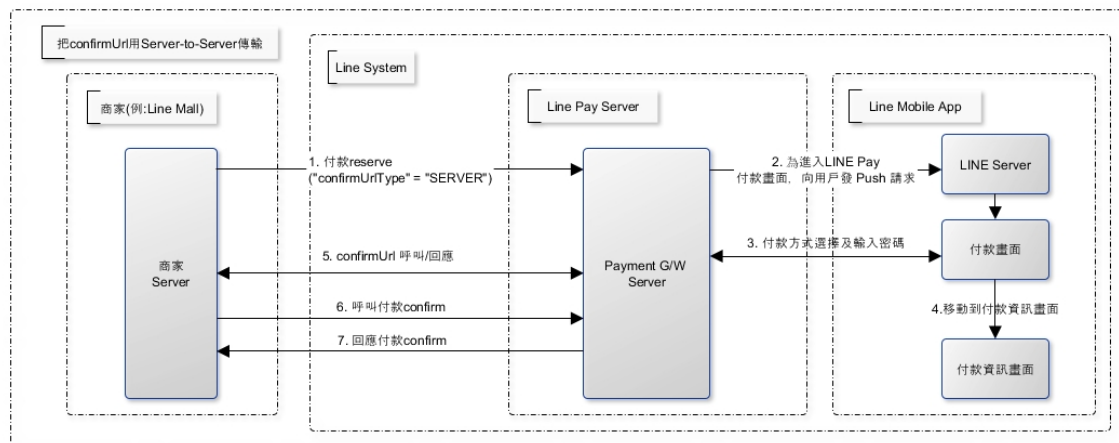


圖 7 伺服器對伺服器呼叫 confirmUrl

- 商家在做付款 reserve之時，把 confirmUrlType設定為： "SERVER"。
- LINE Pay會員進入到 LINE Pay付款畫面後選擇付款工具，再來輸入密碼。
- LINE Pay伺服器會儲存付款資訊，做付款 reserve時再呼叫加盟店傳達的confirmUrl。
[附錄] 需參考“若把confirmUrl 透過Server-to-Server方式傳送”。
- 商家伺服器需要呼叫 confirm API來做完付款動作。若對於 confirmUrl的回覆沒有正常傳達之時，就無法呼叫 confirm API。

若 LINE Pay app提供之 oneTimeKey來進行付款

在 LINE Pay app, 為了支援 offline付款, 生成 oneTimeKey。oneTimeKey就是替代 LINE的會員資訊之號碼, 可以以條碼或二維碼形式做確認。商家可以掃描其代碼獲得 oneTimeKey, 進行付款。

1. 前提條件

- a. oneTimeKey之有效時間：5分鐘 (從使用者接近 oneTimeKey頁面開始計算)
- b. oneTimeKey是與付款 reserve同時被刪除。

2. 連動流程

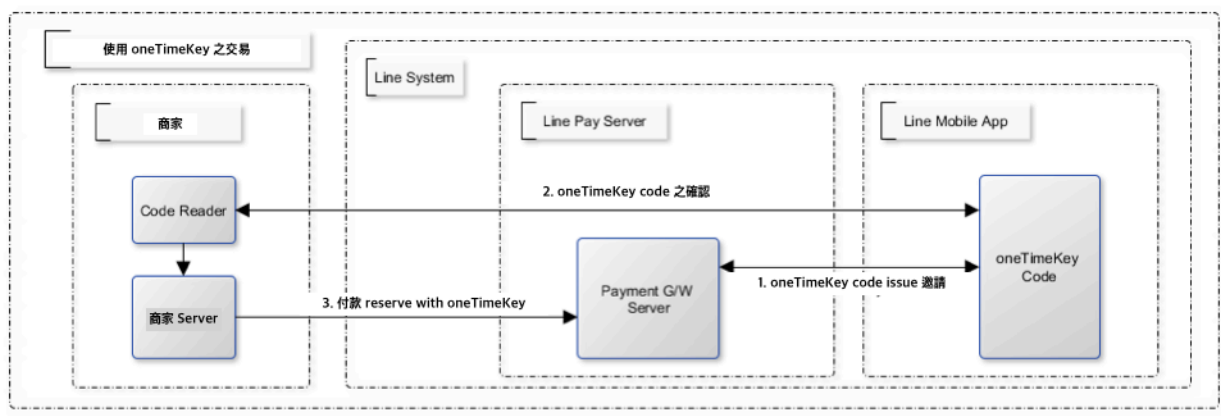


圖 8 oneTimeKey之生成以及 reserve

- a. 商家要讀取 LINE Pay的使用者提供的 oneTimeKey (條碼或二維碼)。
- b. 商家以獲得的 oneTimeKey的資訊來做付款 reserve。
- c. 付款 reserve後, LINE Pay的會員透過手機的 LINE app來確認付款邀請。
- d. LINE Pay的會員進入 LINE Pay付款畫面後選擇付款工具, 再來輸入密碼。
- e. LINE Pay 的會員確認完付款資訊頁面後進行付款時, 按照商家在 reserve API 設定的 confirmUrlType 參數而遇到的付款場景會不同。
 - CLIENT(預設) : confirmUrl 將於買家的瀏覽器打開, 買家移動至該 URL 後, 商家再呼叫 confirm API 來完成付款動作。
 - SERVER : [附錄] 需參考 “若把confirmUrl 透過Server-to-Server方式傳送”。

商家的 Server 呼叫 confirm API 來完成付款流程。若沒有正常收到 confirmUrl 的回覆時, 無法呼叫 Confirm API。

自動付款

付款請求 - 付款畫面

除了 "payType" 必須設定為 "PREAPPROVED" 之外，整個流程與一般付款相同。

付款畫面 - 付款完成與核發 regKey (用於自動付款)

傳遞 regKey，額外附帶付款確認的回應資訊。商家必須儲存此金鑰，稍後才能使用自動付款。

自動付款

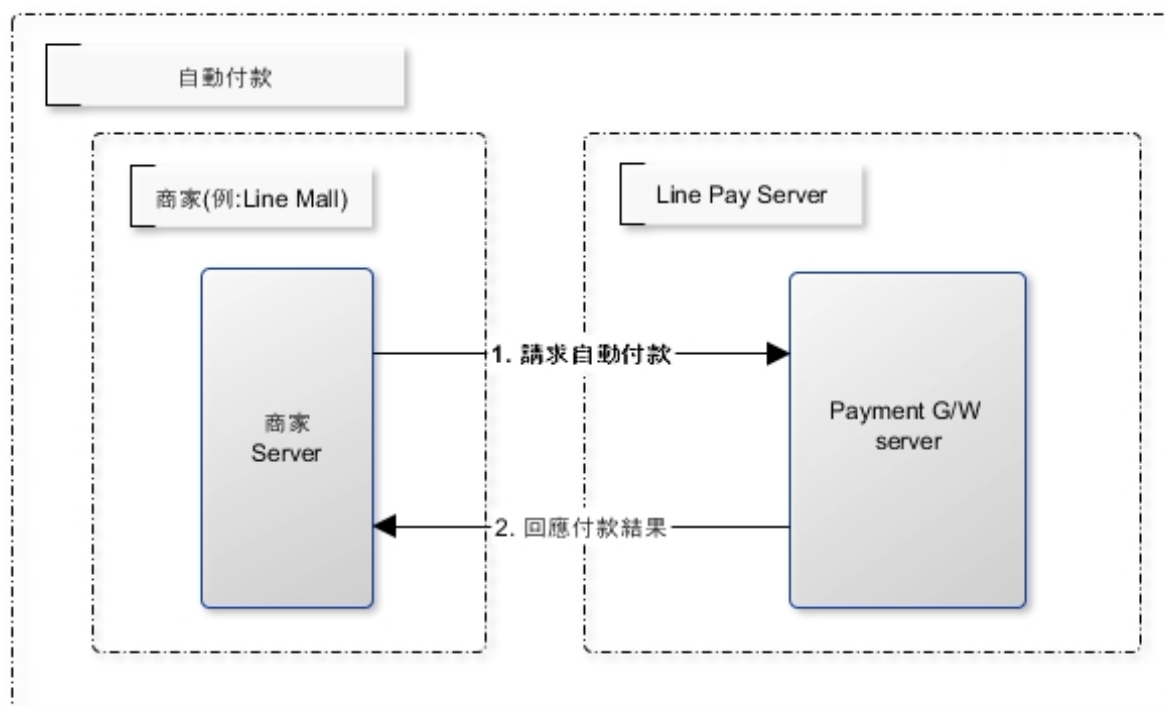


圖 9 自動付款

當付款確認時，商家伺服器使用 regKey 呼叫 "自動付款 API" 來進行付款。LINE Pay 用戶不介入付款流程，但是在付款完成時會接到通知。

regKey (用於自動付款) 的註銷

您可以呼叫註銷 regKey API 使不需使用的 regKey 過期。如需詳細資訊，請參閱「註銷 regKey」。

如何使用 LINE Pay 的各項 API

本章將詳細說明如何實作 LINE Pay 付款。使用 LINE Pay API 之前，可以在評估流程完成後到商家中心²裡檢視商家的驗證資訊。

整合 LINE Pay 時所需的驗證資訊如下：

- 通路 ID
- 通路密鑰

所需的資訊會透過標頭轉送，可以在每個 API 規格中檢視。

如果您對於 LINE Pay 整合有任何問題，或發生內部錯誤，請與技術支援 (pay_tech@linecorp.com) 聯絡。

連動環境

- Beta：只能在LINE 內部網路才能測試。
- Sandbox：為了測試連動的環境。於 web simulation page 模擬 LINE Pay App 的付款交易。
[附錄] 需參考“Sandbox 使用方法”。
- Real：實際服務營運環境

² 商家中心是可讓 LINE Pay 商家在此確認交易細節與入帳紀錄，也可以更新基本商家資訊和提供 1 對 1 服務的管道。商家目前的服務狀態可以從商家基本資訊中得知。

API 變更紀錄

注意

- API 功能多加或變更時, API 版本會跟著升級。
 - 保障舊版本的運作暢通, 不過建議適用最新版本。
-

2015-06-24

全新定義可嘗試重複呼叫之 returnCode

1. API : 退款、授權作廢處理、請款
2. 變更內容 : 多加在 LINE Pay 系統內部發生錯誤時候的 returnCode。 (此時能再次呼叫 API)

2015-04-30

1. API 版本升級為 v2
2. Promotion 有關 value 多家
 - a. API : [付款 confirm](#), [請款](#), [查看授權紀錄](#), [查看付款紀錄](#)
 - b. 變更內容 : API 回覆資訊中, 多加了 “payInfo[].method” value
 - DISCOUNT : 商家要提供給 LINE Pay 客戶折扣優惠時候需要放置傳達的 value
3. Token 為基礎的付款連動支援
 - a. API : [付款 reserve](#)
 - b. 變更內容 : 為了透過 LINE Pay scanner(從 LINE 5.1 版本開始支援) 付款, 多加了 token。
 - paymentAccessToken : 多加在 API 的回傳訊息裡面, 為了客戶把訊息輸入在 LINE Pay scanner 的 “input key code” 時候馬上引他到 LINE Pay 付款畫面之用途。
4. 支援自動付款授權與請款之分開
 - a. API : [自動付款](#)
 - b. 變更內容 : 在自動付款時候, 為了像一般付款一樣可以把授權與請款分開, 多加了參數。
 - capture : 請款與否
 - true(預設): 呼叫付款 confirm API 時, 把授權與請款一起處理
 - false: 呼叫付款 confirm API 時, 先做授權後再需要呼叫「[請款 API](#)」才能完成交易。
5. regKey 支援授權
 - a. API : . [查看 regKey 使用可否用途之 API](#)
 - b. 變更內容 : 查看 regKey 還是否有效之過程裡, 為了試圖往發卡行發送最少金額之授權請求, 多加了確認自動付款可否成立用途之參數。
 - creditCardAuth : 行用卡最少金額授權試圖與否
 - true : 要試圖
 - false(預設) : 在 LINE Pay 內部檢查數據之有效性

付款reserve

在 LINE Pay 中保留付款資訊。

付款reserve API

在使用 LINE Pay 付款前確認是否為正常的商家，然後保留付款所需的資訊。成功付款reserve 後，商家會收到一個「交易編號」，這個鍵值會在付款完成或退款時使用。

付款reserve API 規格

表 1 付款reserve API 規格

項目	說明
Method	POST
Required	Content-Type:application/json; charset=UTF-8
Request Header	X-LINE-ChannelId:{Integration information - channelId} X-LINE-ChannelSecret:{Integration information - channelSecretKey}
Request timeout	connecton: 5秒 read: 20秒
beta	https://lgw-pay.line-apps-beta.com/v2/payments/request
sandbox	https://sandbox-api-pay.line.me/v2/payments/request
Real	https://api-pay.line.me/v2/payments/request

表 2 付款reserve API 請求參數

項目	資料型別	是否必要?	說明
productName	String (4000 位元組)	Y	產品名稱 (charset:"UTF-8")
productImageUrl	String (500 位元組)	Y	產品影像 URL (或使用品牌logo URL) 顯示於付款畫面上的影像 URL <ul style="list-style-type: none"> 大小：84 x 84 (只顯示於付款畫面的影像，建議依照指南中的規範製作)
amount	number ³	Y	付款金額

³ 允許小數位數的數值資料型別

項目	資料型別	是否必要?	說明
currency	String (3 位元組)	Y	付款貨幣 (ISO 4217) 支援的貨幣如下： <ul style="list-style-type: none"> • TWD • JPY • USD • THB
mid	String (50 位元組)	N	LINE 用戶 ID 將要進行付款的 LINE 使用者之 mid
oneTimeKey	String (12byte)	N	oneTimeKey 是讀取 LINE Pay app 所提供之二維碼、條碼後之結果。替代 LINE Pay 會員之 mid。有效時間為5分鐘，與 rserve 同時會被刪除。 LINE Pay app 的 QR/BarCode 是從5.1版本開始支援
confirmUrl	String (500 位元組)	Y	買家在 LINE Pay 選擇付款方式並輸入密碼後，被重新導向到商家的 URL。 <ul style="list-style-type: none"> • 在重新導向的 URL 上，商家可以呼叫付款confirm API 並完成付款。 • LINE Pay 會傳遞額外的 "transactionId" 參數。 • "confirmUrlType" 設為 "SERVER" 來呼叫時，必須使用 "https" 協定，並在 real 環境必須使用 "可信賴之SSL憑證書"。但，在 sandbox 環境是因為測試期間使用的環境，因此可以忽略憑證書錯誤。
參考			
confirmUrl 詳細說明以及例外狀況			
confirmUrlType	String	N	confirmUrl 類型。 買家在 LINE Pay 選擇付款方式並輸入密碼後，被重新導向到的 URL 所屬的類型。 <ul style="list-style-type: none"> • CLIENT: 手機交易流程 (預設) • SERVER: 網站交易流程。用戶只需要查看 LINE Pay 的付款資訊畫面，然後通知商家伺服器可以付款。
checkConfirmUrlBrowser	Boolean	N	User 前往 confirmUrl 移動時，確認使用的瀏覽器相同與否 <ul style="list-style-type: none"> • true: 若買家請求付款的瀏覽器與實際打開 confirmUrl 的瀏覽器不同之時，LINE Pay 將會提供請買家回到原本的瀏覽器的介紹頁面。 • false(預設): 不會確認瀏覽器，立即打開 confirmUrl。
參考			
confirmUrl 詳細說明以及例外狀況			

項目	資料型別	是否必要?	說明
cancelUrl	String (500 位元組)	N	取消付款頁面的 URL <ul style="list-style-type: none"> 當 LINE Pay 用戶取消付款後，從 LINE 應用程式付款畫面重新導向的 URL (取消付款後，透過行動裝置進入商家應用程式或網站的商家 URL)。 商家傳送的 URL 會依現況直接使用。 LINE Pay 不會傳遞任何額外的參數。
packageName	String (4000 位元組)	N	在 Android 各應用程式間轉換時，防止網路釣魚詐騙的資訊。
orderId	String (100 位元組)	Y	商家與該筆付款請求對應的訂單編號 <ul style="list-style-type: none"> 這是商家自行管理的唯一編號。
deliveryPlacePhone	String (100 位元組)	N	收件人的聯絡資訊 (用於風險管理)
payType	String (12 位元組)	N	付款類型 <ul style="list-style-type: none"> NORMAL:單筆付款 (預設) PREAPPROVED:自動付款
langCd	String	N	等待付款畫面 (paymentUrl) 的語言代碼。共支援六種語言。 <ul style="list-style-type: none"> ja:日文 ko:韓文 en:英文 zh-Hans:簡體中文 zh-Hant:繁體中文 th:泰文 <ul style="list-style-type: none"> 語言代碼不是強制性的，如果未收到此代碼，則會根據 accept-language 標頭支援多語言。 如果收到不支援的 langCd，預設會使用英文 ("en")。 BCP-47 格式：http://en.wikipedia.org/wiki/IETF_language_tag
capture	Boolean	N	指定是否請款 <ul style="list-style-type: none"> true:呼叫付款confirm API 時，立即進行付款授權與請款 (預設)。 false:呼叫付款confirm API 時，只有經過授權，然後透過呼叫 "請款 API" 分開請款，才能完成付款。

表 3 付款reserve API 回應

項目	資料型別	說明
returnCode	String (4 位元組)	結果代碼 <ul style="list-style-type: none"> • 0000:成功 • 1104:找不到商家 • 1105:此商家無法使用 LINE Pay • 1106:標頭資訊錯誤 • 1124:金額錯誤 (scale) • 1133: 非有效之 oneTimeKey。 • 1145:正在進行付款 • 1172: 已有同一訂單編號的交易資料 • 1178:不支援的貨幣 • 1194:此商家無法使用自動付款 • 9000:內部錯誤
returnMessage	String	結果訊息或失敗理由。範例如下： <ul style="list-style-type: none"> • 商家未獲授權，無法付款 • 商家驗證資訊錯誤
info.transactionId	Number	交易編號 (19 位數)
info.paymentUrl.web	String	付款請求後所前往的網頁 URL <ul style="list-style-type: none"> • 在網頁交易環境中請求付款時使用 • LINE Pay 等待付款畫面的 URL • 重新導向到提供的 URL，不附帶任何額外參數 • 在桌機打開跳窗時，其大小為：Width: 700px, Height : 546px
info.paymentUrl.app	String	前往付款畫面的應用程式 URL <ul style="list-style-type: none"> • 在機交易請求付款時使用 • 將 URL 從商家手機交易重新導向到 LINE 應用程式 • 有關使用應用程式 URL 所需的範例程式碼，請參考「附錄」PaymentUrl 應用程式使用指南
info.paymentAccessToken	String	在 LINE Pay app 輸入代碼來代替適用掃描器(Scanner)之時候所適用的代碼值(共12位數)。 (LINE Pay app 的掃描器是從 5.1 版本開始支援)

使用付款reserve API 的範例

付款reserve API 請求

```
{
  "productName": "test product",
  "productImageUrl": "http://testst.com",
  "amount": 10,
  "currency": "USD",
  "mid": "os89dufgoiw8yer9021384rdfeq",
  "orderId": "20140101123456789",
  "confirmUrl": "naversearchapp://inappbrowser?url= http%3A%2F%2FtestMall.com%2FcheckResult.nhn%3ForderId%3D20140101123456789",
  "cancelUrl": "naversearchapp://inappbrowser?url= http%3A%2F%2FtestMall.com%2ForderSheet.nhn%3ForderId%3D20140101123456789",
}
```

```
"capture" : "true",  
"confirmUrlType" : "CLIENT"  
}
```

付款reserve API 回應

```
{  
  "returnCode" : "0000",  
  "returnMessage" : "OK",  
  "info" : {  
    "transactionId" : 123123123123,  
    "paymentUrl" : {  
      "web" : "http://web-pay.line.me/web/wait?transactionReserveId=blahblah",  
      "app" : "line://pay/payment/blahblah"  
    },  
    "paymentAccessToken" : "187568751124"  
  }  
}
```

付款confirm

付款confirm API

此 API 可讓商家完成付款。商家必須呼叫付款confirm API，才能實際完成付款。不過，當付款reserve 的 "capture" 參數為 "false" 時，付款狀態會變為 "AUTHORIZATION"，只有在呼叫 "請款 API" 後才能完成付款。

付款confirm API 規格

表 4 付款confirm API 規格

項目	說明
Method	POST
Required	Content-Type:application/json; charset=UTF-8
Request Header	X-LINE-ChannelId:{Integration information - channelId} X-LINE-ChannelSecret:{Integration information - channelSecretKey}
Request timeout	connecton: 5秒 read: 40秒
beta	https://lgw-pay.line-apps-beta.com/v2/payments/{ transactionId }/confirm
sandbox	https://sandbox-api-pay.line.me/v2/payments/{ transactionId }/confirm
Real	https://api-pay.line.me/v2/payments/{ transactionId }/confirm

表 5 付款confirm API 請求參數

項目	資料型別	是否必要？	說明
amount	number	Y	付款金額
currency	String (3 位元組)	Y	付款貨幣 (ISO 4217) 支援的貨幣如下。 <ul style="list-style-type: none"> • TWD • JPY • USD • THB

⁴ 指呼叫付款 reserve API 時回傳的 transactionId。

表 6 付款confirm API 回應

項目	資料型別	說明
returnCode	String (4 位元組)	結果代碼 <ul style="list-style-type: none"> • 0000:成功 • 1101: 買家不是 LINE Pay 會員 • 1102: 買方被停止交易 • 1104: 找不到商家 • 1105: 此商家無法使用 LINE Pay • 1106: 標頭資訊錯誤 • 1110: 無法使用的信用卡 • 1124: 金額錯誤 (scale) • 1141: 付款帳戶狀態錯誤 • 1142: 餘額不足 • 1150: 找不到交易 • 1152: 已有保存付款 • 1153: 付款reserve時的金額與申請的金額不一致 • 1159: 無付款申請資訊 • 1169: 付款confirm所需要資訊錯誤 (在LINE Pay) • 1170: 使用者帳戶的餘額有變動 • 1172: 已有同一訂單編號的交易資料 • 1180: 付款時限已過 • 1198: 正在處理請求... • 1199: 內部請求錯誤 • 1280: 信用卡付款時候發生了臨時錯誤 • 1281: 信用卡付款錯誤 • 1282: 信用卡授權錯誤 • 1283: 因疑似詐騙, 拒絕付款 • 1284: 暫時無法以信用卡付款 • 1285: 信用卡資訊不完整 • 1286: 信用卡付款資訊不正確 • 1287: 信用卡已過期 • 1288: 信用卡的額度不足 • 1289: 超過信用卡付款金額上限 • 1290: 超過一次性付款的額度 • 1291: 此信用卡已被掛失 • 1292: 此信用卡已被停卡 • 1293: 信用卡驗證碼 (CVN) 無效 • 1294: 此信用卡已被列入黑名單 • 1295: 信用卡號無效 • 1296: 無效的金額 • 1298: 信用卡付款遭拒 • 9000: 內部錯誤
returnMessage	String	結果訊息或失敗理由。範例如下： <ul style="list-style-type: none"> • 商家未獲授權, 無法付款 • 商家驗證資訊錯誤
info.orderId	String	商家在付款reserve 時傳送的訂單編號
info.transactionId	number	付款 reserve 後, 做為結果所收到的交易編號

項目	資料型別	說明
info.authorizationExpirationDate	String	可選擇的授權過期日 (ISO 8601) <ul style="list-style-type: none"> 當付款狀態為 AUTHORIZATION (capture=false) 時
info.payInfo[].method	String	使用的付款方式 (信用卡 : CREDIT_CARD, 餘額 : BALANCE, 折扣: DISCOUNT)
info.payInfo[].amount	number	付款金額
info.payInfo[].creditCardNickname	String	(paytype = Preapproved 之時候) 信用卡暱稱 <ul style="list-style-type: none"> 由 LINE Pay 管理的最初登錄的信用卡名稱。 如果 LINE Pay 用戶未設定暱稱, 則顯示空白文字 ("")。 此暱稱可由用戶在 LINE Pay 中請求更改, 更改後的名稱不會告知商家。
info.payInfo[].creditCardBrand	String	(paytype = Preapproved 之時候) 信用卡品牌 <ul style="list-style-type: none"> VISA MASTER AMEX DINERS JCB
info.regKey	String	paytype 為 Preapproved 之時候, 以後可直接選用的自動付款金鑰 (15 位數)

使用付款 confirm API 的範例

付款 confirm API 請求

```
POST https://sandbox-api-pay.line.me/v2/payments/sdhqiwouehrafdasrqoi123as/confirm HTTP/1.1
Content-Type: application/json
{"amount":1000,"currency":"TWD" }
```

付款 confirm API 回應

```
// 若付款類型("payType")為 "NORMAL"時
{
  "returnCode" : "0000",
  "returnMessage" : "OK",
  "info" : {
    "orderId" : "order_210124213",
    "transactionId" : 20140101123123123,
    "payInfo" : [
      {"method" : "BALANCE", "amount" : 10},
      {"method" : "DISCOUNT", "amount" : 10}
    ]
  }
}

// 若付款類型("payType")為 "PREAPPROVED"時
{
  "returnCode" : "0000",
  "returnMessage" : "OK",
  "info" : {
    "orderId" : "order_210124213",
    "transactionId" : 20140101123123123,
```

```
    "payInfo" : [  
      {"method" : "CREDIT_CARD", "amount" : 10, "creditCardNickname" : "test", "creditCardBrand"  
" : "VISA"}  
    ],  
    "regKey" : "RKf930b19XTF1TX"  
  }  
}
```

請款

如果商家呼叫付款reserve API 時 "capture" 為 "false", 則只有呼叫請款 API 後才能完成付款。

請款 API

confirm API 執行結果只有授權的付款進行請款動作。

請款 API 規格

表 7 請款 API 規格

項目	說明
Method	POST
Required	Content-Type:application/json; charset=UTF-8
Request Header	X-LINE-ChannelId:{Integration information - channelId} X-LINE-ChannelSecret:{Integration information - channelSecretKey}
Request timeout	connecton: 5秒 read: 60秒
Beta	https://lgw-pay.line-apps-beta.com/v2/payments/authorizations/{transactionId ⁵ }/capture
Sandbox	https://sandbox-api-pay.line.me/v2/payments/authorizations/{transactionId}/capture
Real	https://api-pay.line.me/v2/payments/authorizations/{transactionId}/capture

表 8 請款 API 請求參數

項目	資料型別	是否必要?	說明
amount	number	Y	付款金額
currency	String (3 位元組)	Y	付款貨幣 (ISO 4217) 支援下列貨幣： <ul style="list-style-type: none"> • TWD • JPY • USD • THB

⁵ 指呼叫付款 reserve API 時回傳的 transactionId。

表 9 請款 API 回應

項目	資料型別	說明
returnCode	String (4 位元組)	<p>結果代碼</p> <ul style="list-style-type: none"> • 0000: 成功 • 1104: 找不到商家 • 1105: 此商家無法使用 LINE Pay • 1106: 標頭資訊錯誤 • 1150: 找不到交易記錄 • 1155: 交易編號不符合退款資格 • 1170: 使用者帳戶的餘額有變動 • 1172: 已有同一訂單編號的交易資料 • 1179: 無法處理狀態 • 1183: 付款金額必須大於 0 • 1184: 付款金額比付款申請時候的金額還大 • 1198: 正在處理請求... • 1199: 內部請求錯誤 • 1280: 信用卡付款時候發生了臨時錯誤 • 1281: 信用卡付款錯誤 • 1282: 信用卡授權錯誤 • 1283: 因疑似詐騙, 拒絕付款 • 1284: 暫時無法以信用卡付款 • 1285: 信用卡資訊不完整 • 1286: 信用卡付款資訊不正確 • 1287: 信用卡已過期 • 1288: 信用卡的額度不足 • 1289: 超過信用卡付款金額上限 • 1290: 超過一次性付款的額度 • 1291: 此信用卡已被掛失 • 1292: 此信用卡已被停卡 • 1293: 信用卡驗證碼 (CVN) 無效 • 1294: 此信用卡已被列入黑名單 • 1295: 信用卡號無效 • 1296: 無效的金額 • 1298: 信用卡付款遭拒 • 1900: 暫時錯誤, 請稍後再試。 • 1901: 暫時錯誤, 請稍候再試。 • 1999: 嘗試呼叫的資訊與前一次的資訊不符。 • 9000: 內部錯誤 <p># 若得到 1900, 1901, 1999之代碼, 可以再試圖重複呼叫。 #若得到 1199, 1280~1298之代碼, 該筆交易會自動被取消。</p>
returnMessage	String	<p>結果訊息或失敗理由。範例如下：</p> <ul style="list-style-type: none"> • 商家未獲授權, 無法付款 • 商家驗證資訊錯誤

項目	資料型別	說明
info.orderId	String	商家在付款reserve 時傳遞的訂單編號
info.transaction number Id		付款 reserve API 回傳之「交易號碼」訊息(共19位數)
info.payInfo[].method	String	付款方式 (信用卡：CREDIT_CARD, 餘額：BALANCE, 折扣: DISCOUNT)
info.payInfo[].amount	number	付款金額

使用請款 API 的範例

請款 API 請求

```
POST https://sandbox-api-pay.line.me /v2/payments/authorizations/sdhqiwouehrafdasrqoi123as/capture
HTTP/1.1
Content-Type: application/json
{"amount":1000,"currency":"TWD"}
```

請款 API 回應

```
{
  "returnCode" : "0000",
  "returnMessage" : "OK",
  "info" : {
    "orderId" : "order_210124213",
    "transactionId" : 20140101123123123,
    "payInfo" : [
      {"method" : "BALANCE", "amount" : 10},
      {"method" : "CREDIT_CARD", "amount" : 10}
    ]
  }
}
```

授權作廢

將已授權的交易作廢。

授權作廢 API

將先前已授權的付款作廢。已經請款的付款可以藉由使用「退款 API」進行退款。

授權作廢 API 規格

表 10 授權作廢 API 規格

項目	說明
Method	POST
Required	Content-Type:application/json; charset=UTF-8
Request Header	X-LINE-ChannelId:{Integration information - channelId} X-LINE-ChannelSecret:{Integration information - channelSecretKey}
Request timeout	connecton: 5秒 read: 20秒
Beta	https://lgw-pay.line-apps-beta.com/v2/payments/authorizations/{transactionId ⁶ }/void
Sandbox	https://sandbox-api-pay.line.me/v2/payments/authorizations/{transactionId}/void
Real	https://api-pay.line.me/v2/payments/authorizations/{transactionId}/void

表 11 授權作廢 API 回應

項目	資料型別	說明
----	------	----

⁶ 指呼叫付款 reserve API 時回傳的 transactionId。

項目	資料型別	說明
returnCode	String (4 位元組)	結果代碼 <ul style="list-style-type: none"> • 0000: 成功 • 1101: 買家不是 LINE Pay 會員 • 1102: 買方被停止交易 • 1104: 找不到商家 • 1105: 此商家無法使用 LINE Pay • 1106: 標頭資訊錯誤 • 1150: 找不到交易記錄 • 1155: 交易編號不符合退款資格 • 1165: 交易已進行退款 • 1170: 使用者帳戶的餘額有變動 • 1198: 正在處理請求... • 1199: 內部請求錯誤 • 1900: 暫時錯誤, 請稍後再試。 • 1903: 暫時錯誤, 請稍候再試。 • 1999: 嘗試呼叫的資訊與前一次的資訊不符。 • 9000: 內部錯誤 # 若得到 1900, 1903, 1999 之代碼, 可以再試圖重複呼叫。
returnMessage	String	結果訊息或失敗理由。範例如下： <ul style="list-style-type: none"> • 商家未獲授權, 無法付款 • 商家驗證資訊錯誤

使用授權作廢 API 的範例

授權作廢 API 請求

```
POST https://sandbox-api-pay.line.me /v2/payments/authorizations/sdhqiwouehrafdasrqoi123as/void HTTP/1.1
Content-Type: application/json
```

授權作廢 API 回應

```
{
  "returnCode" : "0000",
  "returnMessage" : "OK",
}
```

查看授權記錄

取得查看授權記錄 API

取得透過 LINE Pay 授權項目的詳細說明。此 API 只會取得已經授權或授權已經作廢的資料；已經請款的項目則可使用「取得查看付款紀錄 API」來檢視。

取得查看授權記錄 API 規格

表 12 檢視查看授權記錄API規格

項目	說明
Method	GET
Required	X-LINE-ChannelId:{Integration information – channelId}
Request Header	X-LINE-ChannelSecret:{Integration information – channelSecretKey}
Request timeout	connecton: 5秒 read: 20秒
beta	https://lgw-pay.line-apps-beta.com/v2/payments/authorizations
sandbox	https://sandbox-api-pay.line.me/v2/payments/authorizations
Real	https://api-pay.line.me/v2/payments/authorizations

表 13 取得查看授權記錄 API 參數

項目	資料型別	是否必要？	說明
transactionId	number	N	由 LINE Pay 核發的交易編號
orderId	String	N	商家的訂單編號

* 呼叫此 API 時，需要傳遞一個以上的參數。您最多可以查看 100 筆交易的詳細說明。

表 14 取得查看授權記錄 API 回應

項目	資料型別	說明
returnCode	String (4 位元組)	結果代碼 <ul style="list-style-type: none"> • 0000: 成功 • 1104: 找不到商家 • 1105: 此商家無法使用 LINE Pay • 1106: 標頭資訊錯誤 • 1150: 找不到交易記錄 • 1177: 超過允許擷取的交易數目 (100) • 9000: 內部錯誤
returnMessage	String	結果訊息或失敗理由
info[].transactionId	number	交易編號 (19 位數)
info[].transactionDate	String	交易日期 (ISO 8601)
info[].transactionType	String	交易類型 <ul style="list-style-type: none"> • PAYMENT:付款 • PAYMENT_REFUND:退款 • PARTIAL_REFUND:部分退款
info[].payStatus	String	付款狀態 <ul style="list-style-type: none"> • AUTHORIZATION:已授權 • VOIDED_AUTHORIZATION:授權已作廢 (呼叫了「授權作廢 API」) • EXPIRED_AUTHORIZATION:過期的授權 (LINE Pay 允許的授權期限已過期)
info[].payInfo[].method	String	付款方式 (信用卡：CREDIT_CARD, 餘額：BALANCE, 折扣：DISCOUNT)
info[].payInfo[].amount	number	交易金額 (建立交易編號時) 最終交易金額為 $\text{sum}(\text{info}[\text{[]}].\text{payInfo}[\text{[]}].\text{amount}) - \text{sum}(\text{refundList}[\text{[]}].\text{refundAmount})$ 。
info[].productName	String	產品名稱
info[].currency	String	貨幣 (ISO 4217)
info[].orderId	String	商家的訂單編號
info[].authorizationExpireDate	String	授權過期日期與時間 (ISO 8601)

使用取得查看授權記錄 API 的範例

取得查看授權記錄 API 請求

```
GET https://sandbox-api-pay.line.me /v2/payments/authorizations?transactionId=2014123123123&transactionId=201411111122222 HTTP/1.1
```

取得查看授權記錄 API 回應

```
{
  "returnCode": "0000",
```

```
"returnMessage":"success",
"info":{
  "transactionId":1020140728100001997,
  "transactionDate":"2014-07-28T09:48:43Z",
  "transactionType":"PAYMENT",
  "payInfo":[
    {"method":"BALANCE", "amount":10},
    {"method":"CREDIT_CARD", "amount":10}
  ],
  "productName":"tes production",
  "currency":"USD",
  "orderId":"20140101123123123",
  "payStatus" : "AUTHORIZATION",
  "authorizationExpireDate" : "2014-07-28T09:48:43Z"
}
}
```

查看付款紀錄

取得查看付款紀錄 API

取得透過 LINE Pay 進行付款的詳細說明。此 API 只會取得已經請款的付款。

取得查看付款紀錄 API 規格

表 15 檢視查看付款紀錄 API 規格

項目	說明
Method	GET
Required	X-LINE-ChannelId:{Integration information - channelId}
Request Header	X-LINE-ChannelSecret:{Integration information - channelSecretKey}
Request timeout	connecton: 5秒 read: 20秒
beta	https://lgw-pay.line-apps-beta.com/v2/payments
sandbox	https://sandbox-api-pay.line.me/v2/payments
Real	https://api-pay.line.me/v2/payments

表 16 取得查看付款紀錄 API 參數

項目	資料型別	是否必要?	說明
transactionId	number	N	由 LINE Pay 核發的交易編號, 用於付款或退款
orderId	String	N	商家的訂單編號

* 呼叫此 API 時, 需要傳遞一個以上的參數。您最多可以查看 100 筆交易的詳細說明。

表 17 取得查看付款紀錄 API 回應

項目	資料型別	說明
returnCode	String (4 位元組)	結果代碼 <ul style="list-style-type: none"> 0000: 成功 1104: 找不到商家 1105: 此商家無法使用 LINE Pay 1106: 標頭資訊錯誤 1150: 找不到交易記錄 1177: 超過允許擷取的交易數目(100) 9000 : 內部錯誤
returnMessage	String	結果訊息或失敗理由
info[].transactionId	number	交易編號 (19 位數)
info[].transactionDate	String	交易日期與時間 (ISO 8601)
info[].transactionType	String	交易類型 <ul style="list-style-type: none"> PAYMENT:付款 PAYMENT_REFUND:退款 PARTIAL_REFUND:部分退款
info[].payInfo[].method	String	使用的付款方式 (信用卡 : CREDIT_CARD, 餘額 : BALANCE, 折扣: DISCOUNT)
info[].payInfo[].amount	number	交易金額 (產生交易編號時的交易金額) 擷取原始交易時的最終交易金額為 $\text{sum}(\text{info}[\text{].payInfo}[\text{].amount) - \text{sum}(\text{refundList}[\text{].refundAmount)}$ 。
info[].productName	String	產品名稱
info[].currency	String	貨幣 (ISO 4217)
info[].orderId	String	商家的訂單編號

擷取原始交易與發生退款時

info[].refundList[].refundTransactionId	number	退款的交易編號 (19 位數)
info[].refundList[].transactionType	String	交易類型 <ul style="list-style-type: none"> PAYMENT_REFUND:退款 PARTIAL_REFUND:部分退款

項目	資料型別	說明
info[].refundList[].refund number Amount		退款金額
info[].refundList[].refund String TransactionDate		退款的交易日期與時間 (ISO 8601)
擷取退款時		
info[].originalTransaction number Id		原始付款的交易編號 (19 位數)

使用取得查看付款紀錄 API 的範例

取得查看付款紀錄 API 請求

```
GET https://sandbox-api-pay.line.me /v2/payments?orderId=20140101123123123&orderId=2014111111222222 HTTP/1.1
```

取得查看付款紀錄 API 回應 (擷取原始交易時)

```
{
  "returnCode":"0000",
  "returnMessage":"success",
  "info":[{
    "transactionId":1020140728100001997,
    "transactionDate":"2014-07-28T09:48:43Z",
    "transactionType":"PAYMENT",
    "payInfo":[
      {"method":"BALANCE", "amount":10},
      {"method":"CREDIT_CARD", "amount":10}
    ],
    "productName":"tes production",
    "currency":"USD",
    "orderId":"20140101123123123",
    "refundList":[
      {
        "refundTransactionId":20213901923123,
        "transactionType":"PARTIAL_REFUND",
        "refundAmount":-1,
        "refundTransactionDate":"2014-07-28T09:48:52Z"
      },
      {
        "refundTransactionId":20213901923123,
        "transactionType":"PARTIAL_REFUND",
        "refundAmount":-1,
        "refundTransactionDate ":"2014-07-28T09:49:28Z"
      },
      {
        "refundTransactionId":20213901923123,
        "transactionType":"PARTIAL_REFUND",
        "refundAmount":-1,
        "refundTransactionDate ":"2014-07-28T09:53:18Z"
      }
    ]
  }
]
```

取得查看付款紀錄 API 回應 (擷取退款時)

```
{
```

```
"returnCode":"0000",
"returnMessage":"success",
"info":[{"
  "transactionId":1020140728100001997,
  "transactionDate":"2014-07-28T09:48:43Z",
  "transactionType":"PARTIAL_REFUND",
  "amount":-5,
  "productName": "",
  "currency":"USD",
  "orderId":"20140101123123123",
  "originalTransactionId":1020140728100001999
}]
}
```

已付款項目的退款

退款 API

請求對 LINE Pay 付款完成的項目進行退款。退款時必須提供 LINE Pay 用戶的付款交易編號，也可以視退款金額進行部分退款。

退款 API 規格

表 18 退款 API 規格

項目	說明
method	POST
Required	Content-Type:application/json; charset=UTF-8
Request Header	X-LINE-ChannelId:{Integration information - channelId} X-LINE-ChannelSecret:{Integration information - channelSecretKey}
Request timeout	connecton: 5秒 read: 20秒
beta	https://lgw-pay.line-apps-beta.com/v2/payments/{transactionId}/refund
sandbox	https://sandbox-api-pay.line.me/v2/payments/{transactionId}/refund
Real	https://api-pay.line.me/v2/payments/{transactionId}/refund

* transactionId 所指的是付款交易編號。

表 19 退款 API 參數

項目	資料型別	是否必要?	說明
refundAmount	number	N	退款金額 - 如果未傳遞此參數，則全額退款

表 20 退款 API 回應

項目	資料型別	說明
returnCode	String (4 位元組)	結果代碼 <ul style="list-style-type: none"> 0000: 成功 1101: 買家不是 LINE Pay 會員 1102: 買方被停止交易 1104: 找不到商家 1105: 此商家無法使用 LINE Pay 1106: 標頭資訊錯誤 1124: 金額錯誤 (scale) 1150: 找不到交易記錄 1155: 交易編號不符合退款資格 1163: 因超過退款期限, 無法退款 1164: 超過退款額度 1165: 交易已進行退款 1179: 無法處理狀態 1198: 正在處理請求... 1199: 內部請求錯誤 1900: 暫時錯誤, 請稍後再試。 1902: 暫時錯誤, 請稍候再試。 1999: 嘗試呼叫的資訊與前一次的資訊不符。 9000: 內部錯誤 <p># 若得到 1900, 1902, 1999 之代碼, 可以再試圖重複呼叫。</p>
returnMessage	String	結果訊息或失敗理由
info.refundTransactionId	number	退款的交易編號 (新核發的編號 - 19 位數)
info.refundTransactionDate	String	退款的交易日期與時間 (ISO 8601)

使用退款 API 的範例

退款 API 請求

```
// 若部分退款時
{
  "refundAmount" :1000
}

// 若全額退款時
{}
```

退款 API 回應

```
{
  "returnCode" : "0000",
  "returnMessage" : "OK",
  "info" : {
    "refundTransactionId" : 123123123123,
    "refundTransactionDate" : "2014-01-01T06:17:41Z",
```

```
}  
}
```

查看 regKey 狀態

查看 regKey 狀態 API

使用自動付款 API 前，檢查 regKey 是否存在。

查看 regKey 狀態 API 規格

表 21 查看 regKey 狀態 API 規格

項目	說明
Method	GET
Required	Content-Type:application/json; charset=UTF-8
Request Header	X-LINE-ChannelId:{Integration information – channelId} X-LINE-ChannelSecret:{Integration information – channelSecretKey}
Request timeout	connecton: 5秒 read: 20秒
Beta	https://lgw-pay.line-apps-beta.com/v2/payments/preapprovedPay/{regKey}/check
Sandbox	https://sandbox-api-pay.line.me/v2/payments/preapprovedPay/{regKey}/check
Real	https://api-pay.line.me/v2/payments/preapprovedPay/{regKey}/check

表 22 查看是否可使用 regKey 之 API Query 參數

項目	資料型別	是否必要?	說明
creditCardAuth	Boolean	N	試圖驗證買家在 regKey 設定之信用卡之最少金額與否 <ul style="list-style-type: none"> • true : <ul style="list-style-type: none"> - 驗證 LINE Pay 內部檔案之有效性 - 驗證信用卡最少金額付款 - 在商家資訊激活「最少金額驗證功能」才可適用，且需要 LINE Pay 管理者之檢驗。 • false(預設) : <ul style="list-style-type: none"> - 只驗證 LINE Pay 內部檔案之有效性

表 23 查看 regKey 狀態回應

項目	資料型別	說明
returnCode	String(4 位元組)	傳回代碼 <ul style="list-style-type: none"> • 0000: 成功 • 1101:買家不是 LINE Pay 會員 • 1102: 買方被停止交易 • 1104: 找不到商家 • 1105: 此商家無法使用 LINE Pay • 1106: 標頭資訊錯誤 • 1141: 無法驗證在 regKey 設定之信用卡 • 1154: 自動付款帳戶無法使用 • 1190: regKey 不存在 • 1193: regKey 已過期
returnMessage	String	結果訊息或失敗理由

使用查看 regKey 狀態 API 的範例

查看 regKey 狀態 API 請求

```
GET https://sandbox-api-pay.line.me/v2/payments/preapprovedPay/RK7862119XTF1TT/check HTTP/1.1
X-Line-ChannelId: 1414228945
X-Line-ChannelSecret: 15cf4348793ea65cae10959b7a29fe27
Content-Type: application/json
```

查看 regKey 狀態 API 回應

```
{
  "returnCode" : "0000",
  "returnMessage" : "OK"
}
```

自動付款

自動付款 API

當付款reserve API 的付款類型設定為 PREAPPROVED 時，regKey 會隨付款結果傳回。自動付款 API 會使用 regKey 直接完成付款，無需使用 LINE 應用程式。

自動付款 API 規格

表 24 自動付款 API 規格

項目	說明
Method	POST
Required	Content-Type:application/json; charset=UTF-8
Request Header	X-LINE-ChannelId:{Integration information – channelId} X-LINE-ChannelSecret:{Integration information – channelSecretKey}
Request timeout	connecton: 5秒 read: 40秒
Beta	https://lgw-pay.line-apps-beta.com/v2/payments/preapprovedPay/{regKey}/payment
Sandbox	https://sandbox-api-pay.line.me/v2/payments/preapprovedPay/{regKey}/payment
Real	https://api-pay.line.me/v2/payments/preapprovedPay/{regKey}/payment

表 25 自動付款 API 參數

項目	資料型別	是否必要？	說明
productName	String (4000 位元組)	Y	產品名稱
amount	number	Y	付款金額
currency	String (3 位元組)	Y	付款貨幣 (ISO 4217) 支援的貨幣如下： <ul style="list-style-type: none"> • TWD • JPY • USD • THB
orderId	String (100位元組)	Y	商家與該筆付款請求對應的訂單編號。 這是商家自行管理的唯一編號。
capture	Boolean	N	請款與否 <ul style="list-style-type: none"> • true(預設): 一次處理授權與請款 • false: 處理到授權後需要呼叫「請款 API」才能完成交易

表 26 自動付款 API 回應

項目	資料型別	說明
returnCode	String (4 位元組)	結果代碼 <ul style="list-style-type: none"> • 0000: 成功 • 1101: 買家不是 LINE Pay 會員 • 1102: 買方被停止交易 • 1104: 找不到商家 • 1105: 此商家無法使用 LINE Pay • 1106: 標頭資訊錯誤 • 1110: 無法使用的信用卡 • 1124: 金額錯誤 (scale) • 1141: 付款帳戶狀態錯誤 • 1154: 自動付款帳戶無法使用 • 1172: 已有同一訂單編號的交易紀錄 • 1190: regKey 不存在 • 1193: regKey 已過期 • 1198: 正在處理請求... • 1199: 內部請求錯誤 • 1280: 信用卡付款時候發生了臨時錯誤 • 1281: 信用卡付款錯誤 • 1282: 信用卡授權錯誤 • 1283: 因疑似詐騙, 拒絕付款 • 1284: 暫時無法以信用卡付款 • 1285: 信用卡資訊不完整 • 1286: 信用卡付款資訊不正確 • 1287: 信用卡已過期 • 1288: 信用卡的額度不足 • 1289: 超過信用卡付款金額上限 • 1290: 超過一次性付款的額度 • 1291: 此信用卡已被掛失 • 1292: 此信用卡已被停卡 • 1293: 信用卡驗證碼 (CVN) 無效 • 1294: 此信用卡已被列入黑名單 • 1295: 信用卡號無效 • 1296: 無效的金額 • 1298: 信用卡付款遭拒 • 9000: 內部錯誤 # 當出現錯誤代碼 1280 - 1287, 1290 - 1294 時, 將會註銷 regKey。
returnMessage	String	結果訊息或失敗理由
info.transactionId	number	交易編號 (19 位數)
info.transactionDate	String	交易日期與時間 (ISO 8601)

項目	資料型別	說明
info.authorizationExpirationDate	String	授權作廢日期 (ISO 8601) <ul style="list-style-type: none">● 只有做授權(capture=false)時才要傳

使用自動付款 API 的範例

自動付款 API 請求

```
{
  "productName" : "LINE Music 每月定期",
  "amount" : 10000,
  "currency" : "JPY",
  "orderId" : "testOrd2014121200000001"
}
```

自動付款 API 回應

```
{
  "returnCode" : "0000",
  "returnMessage" : "OK",
  "info" : {
    "transactionId" : 123123123123,
    "transactionDate" : "2014-01-01T06:17:41Z"
  }
}
```

註銷 regKey

註銷 regKey API

註銷為自動付款登錄的 regKey 資訊。一旦呼叫此 API，現有的 regKey 將無法繼續用於自動付款。

註銷 regKey API 規格

表 27 regKey 註銷 API 規格

項目	說明
Method	POST
Required	Content-Type:application/json; charset=UTF-8
Request Header	X-LINE-ChannelId:{Integration information – channelId} X-LINE-ChannelSecret:{Integration information – channelSecretKey}
Request timeout	connecton: 5秒 read: 20秒
Beta	https://lgw-pay.line-apps-beta.com/v2/payments/preapprovedPay/{regKey}/expire
Sandbox	https://sandbox-api-pay.line.me/v2/payments/preapprovedPay/{regKey}/expire
Real	https://api-pay.line.me/v2/payments/preapprovedPay/{regKey}/expire

表 28 註銷 regKey API 回應

項目	資料型別	說明
returnCode	String (4 位元組)	結果代碼 <ul style="list-style-type: none"> • 0000: 成功 • 1104: 找不到商家 • 1105: 此商家無法使用 LINE Pay • 1106: 標頭資訊錯誤 • 1190: regKey 不存在 • 1193: regKey 已過期 • 9000: 內部錯誤
returnMessage	String	結果訊息或失敗理由

使用註銷 regKey API 的範例

註銷 regKey API 請求

```
POST https://sandbox-api-pay.line.me/v2/payments/preapprovedPay/RK123asd213/expire HTTP/1.1
Content-Type: application/json
```

註銷 regKey API 回應

```
{
  "returnCode" : "0000",
```

```
"returnMessage" : "OK",  
}
```

附錄

LINE Pay 錯誤代碼

本節定義了 LINE Pay 中的錯誤代碼。錯誤代碼的 returnMessage 以英文呈現，如果沒有訊息，則顯示連字號 (-)。

表 29 LINE Pay 錯誤代碼

代碼 (returnCode)	說明 (300 位元組)
1101	買家不是 LINE Pay 會員
1102	買方被停止交易
1104	找不到商家
1105	此商家無法使用 LINE Pay
1106	標頭資訊錯誤
1110	無法使用的信用卡
1124	金額錯誤 (scale)
1133	非有效之 oneTimeKey
1141	付款帳戶狀態錯誤
1142	餘額不足
1145	正在進行付款
1150	找不到交易記錄
1152	已有既存付款
1153	付款reserve時的金額與申請的金額不一致
1154	買家設定為自動付款的信用卡變無法使用狀態
1155	交易編號不符合退款資格
1159	無付款申請資訊
1163	可退款日期已過無法退款
1164	超過退款額度
1165	交易已進行退款
1169	付款confirm所需要資訊錯誤 (在LINE Pay)
1170	使用者帳戶的餘額有變動
1172	已有同一訂單編號的交易履歷
1177	超過允許擷取的交易數目 (100)
1178	不支援的貨幣
1179	無法處理狀態
1180	付款時限已過
1183	付款金額必須大於 0

代碼 (returnCode)	說明 (300 位元組)
1184	付款金額比付款申請時候的金額還大
1190	regKey 不存在
1193	regKey 已過期
1194	此商家無法使用自動付款
1197	已在處理使用 regKey 進行的付款
1198	正在處理請求…
1199	內部請求錯誤
1280	信用卡付款時候發生了臨時錯誤
1281	信用卡付款錯誤
1282	信用卡授權錯誤
1283	因疑似詐騙, 拒絕付款
1284	暫時無法以信用卡付款
1285	信用卡資訊不完整
1286	信用卡付款資訊不正確
1287	信用卡已過期
1288	信用卡的額度不足
1289	超過信用卡付款金額上限
1290	超過一次性付款的額度
1291	此信用卡已被掛失
1292	此信用卡已被停卡
1293	信用卡驗證碼 (CVN) 無效
1294	此信用卡已被列入黑名單
1295	信用卡號無效
1296	無效的金額
1298	信用卡付款遭拒
1900	暫時錯誤, 請稍候再試。
1901	暫時錯誤, 請稍候再試。
1902	暫時錯誤, 請稍候再試。
1903	暫時錯誤, 請稍候再試。
1999	嘗試呼叫的資訊與前一次的資訊不符。
2101	參數錯誤
2102	JSON 資料格式錯誤
2103	錯誤請求。請確認 returnMessage。
2104	錯誤請求。請確認 returnMessage。
9000	內部錯誤

如果發生內部錯誤 “returnCode:1199”, 請與技術支援 (pay_tech@linecorp.com) 聯絡。

表 30 每API別 ReturnCode總表

錯誤代碼	reserve	confirm	請款	授權作廢	退款	授權履歷	付款紀錄	自動付款	查看 regKey	註銷 regKey
1101		v		v	v			v	v	
1102		v		v	v			v	v	
1104	v	v	v	v	v	v	v	v	v	v
1105	v	v	v	v	v	v	v	v	v	v
1106	v	v	v	v	v	v	v	v	v	v
1110		v						v		
1124	v	v			v			v		
1133	v								v	
1141		v						v		
1142		v								
1145	v									
1150			v	v	v	v	v			
1152		v								
1153		v								
1154								v	v	
1155			v	v	v					
1159		v								
1163					v					
1164					v					
1165				v						
1169		v								
1170		v	v	v						
1172	v	v	v					v		
1177						v	v			
1178	v									
1179			v	v	v					
1180		v								
1183	v		v					v		
1184			v							
1190								v	v	v
1193								v	v	
1194	v									
1198		v	v	v	v			v		
1199		v	v	v	v			v		
1900			v	v	v					

錯誤代碼	reserve	confirm	請款	授權作廢	退款	授權履歷	付款紀錄	自動付款	查看 regKey	註銷 regKey
1901			v							
1902					v					
1903				v						
1999			v	v	v					
2101	v	v	v	v	v	v	v	v	v	v
2102	v	v	v	v	v	v	v	v	v	v
2103	v	v	v	v	v	v	v	v	v	v
2104	v	v	v	v	v	v	v	v	v	v
9000	v	v	v	v	v	v	v	v	v	v

paymentUrl 應用程式使用指南

本節提供在商家應用程式請求付款後，重新導向至 LINE 應用程式的用法說明。

對於 Android 應用程式

使用以下範例程式碼檢查是否已安裝 LINE 應用程式，以及可用的 LINE Pay 版本。如果已經安裝 LINE 應用程式，且已檢查可用的 LINE Pay 版本，則將用戶重新導向到 LINE Pay 付款畫面。

```
int linePaySupportedVersion = 230;
String paymentUrl = "..."; // This is "paymentUrl.app" URL String.

Context context = getActivity();
try {
    PackageManager pm = context.getPackageManager();
    PackageInfo packageInfo = pm.getPackageInfo("jp.naver.line.android", 0);
    int versionCode = packageInfo.versionCode;
    if (linePaySupportedVersion <= versionCode) {
        launchUri(paymentUrl);
    } else {
        confirmLineInstall(context);
    }
} catch (NameNotFoundException e) {
    confirmLineInstall(context);
}

private void confirmLineInstall(Context context) {
    new AlertDialog.Builder(context)
        .setTitle("LINE Pay")
        .setMessage(getString(R.String.linepay_confirm))
        .setCancelable(false)
        .setPositiveButton(getString(R.String.linepay_install), new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                launchUri("market://details?id=jp.naver.line.android");
            }
        })
        .setNegativeButton(getString(R.String.linepay_cancel), new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
            }
        })
        .show();
}

private void launchUri(String uriString) {
    Uri uri = Uri.parse(uriString);
    Intent intent = new Intent(Intent.ACTION_VIEW, uri);
    startActivity(intent);
}
```

File : res/values/Strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    ...
    <String name="linepay_confirm">Supported by Android/iPhone LINE versions 4.8.0 or higher.</String>
    <String name="linepay_install">Get it now</String>
    <String name="linepay_cancel">cancel</String>
    ...
</resources>
```


對於 iOS 應用程式

使用以下範例程式碼檢查是否已安裝 LINE 應用程式。如果已經安裝 LINE 應用程式，則將用戶重新導向到 LINE Pay 付款畫面。

```
NSString* lineScheme = @"line://";
BOOL installed = [[UIApplication sharedApplication]
                 canOpenURL:[NSURL URLWithString:lineScheme]];
if (installed) {
    UIAlertView *alert =
        [[UIAlertView alloc] initWithTitle:@"LINE Pay" message:NSLocalizedString(@"linepay.confirm", nil)
                                     delegate:self cancelButtonTitle:NSLocalizedString(@"linepay.ok", nil) otherButtonTitles:nil];
    alert.tag = 1;
    [alert show];
} else {
    UIAlertView *alert =
        [[UIAlertView alloc] initWithTitle:@"LINE Pay" message:NSLocalizedString(@"linepay.confirm", nil)
                                     delegate:self cancelButtonTitle:NSLocalizedString(@"linepay.cancel", nil)
                                     otherButtonTitles:NSLocalizedString(@"linepay.install", nil), nil];

    alert.tag = 2;
    [alert show];
}

- (void)alertView:(UIAlertView*)alertView clickedButtonAtIndex:(NSInteger)buttonIndex {
    if (alertView.tag == 1 && buttonIndex == 0) {
        NSString *paymentUrl = ...; // This is "paymentUrl.app" URL String.
        [self launchUrl:paymentUrl];
    } else if (alertView.tag == 2 && buttonIndex == 1) {
        [self launchUrl:@"itms-apps://itunes.apple.com/WebObjects/MZStore.woa/wa/viewSoftware?id=443904275&mt=8"];
    }
}

- (void)launchUrl:(NSString*)urlString {
    NSURL *url = [NSURL URLWithString:urlString];
    [[UIApplication sharedApplication] openURL:url];
}
```

File : en.lproj/Localized.Strings

```
"linepay.confirm" = "Supported by Android/iPhone LINE versions 4.8.0 or higher.";
"linepay.ok" = "OK";
"linepay.cancel" = "Cancel";
"linepay.install" = "Get it now";
```

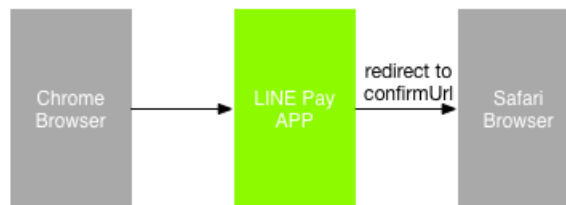
confirmUrl 詳細說明以及例外狀況

confirmUrl 詳細說明

1. 是指使用者在 LINE Pay app 做完付款驗證動作(確認付款工具以及付款密碼)後回去的 callback url。
2. 此 url 並不代表付款動作之完成，呼叫 LINE Pay 所提供的 confirm API 後，最後完成付款動作之商家端的 url。

confirmUrl 例外狀況

：在 LINE Pay，經過使用者的付款驗證後呼叫 confirmUrl 時，隨著使用者的使用環境，有可能發生要求付款的出發 Browser 與收到 confirmUrl 的 Browser 不同之狀況。



付款 reserve API 的 “CheckConfirmUriBrowser” 是 “true” 時

LINE Pay 會確認 browser 啟動付款的 browser 與移動的 browser 是否一致。由起判斷之結果，

- 若一致：移動到 confirmUrl。
- 若不一致：對使用者介紹並要求移動到啟動付款的 browser。

付款 reserve API 的 “CheckConfirmUriBrowser” 是 “false(預設值)” 時

若商家有基於 Browser 之 session data(例：登入 session) 處理之內容，呼叫 confirmUrl 時有可能發生使用者的付款無法正常進行。因此，商家連動 LINE Pay 付款時，必須要如下面去呈現。

1. 呼叫 confirmUrl 時，不履行基於 session data 的驗證，confirmUrl 包含的資訊來完成付款動作。
2. 之後如何處理 session data 之例子
 - a. 若能確認 session data，露出原本商家的付款結果畫面
 - b. 若無法確認 session data，以下兩種方式中選一來呈現。
 - 再生成 session data (例：登入) 後露出付款結果畫面
 - 對使用者介紹並要求回到啟動付款的 browser

伺服器對伺服器呼叫 confirmUrl 時

本節提供從 LINE Pay 伺服器對商家伺服器呼叫 confirmUrl 的規格與範例。

若商家自己管理 ACL(Access Control List)之時候

商家需要登記並管理 LINE Pay server 之 IP。LINE Pay 在每個環境下的 IP 資訊為如下。

- Sandbox :182.162.196.200
- Real : 211.249.40.1~211.249.40.30

表 31 confirmUrl API 規格

項目	說明
Protocol	HTTPS
Method	GET
Timeout	連線：5 秒 讀取：20 秒

confirmUrl 規格

表 32 confirmUrl 參數

參數	說明
orderId	商家在付款reserve 時傳遞的訂單編號
transactionId	預約交易後，做為結果傳回的交易編號

商家不回應對 confirmUrl 的呼叫，因此 LINE Pay 透過檢查 HTTP 狀態代碼來判斷是否成功。如果狀態代碼不是 "200 OK"，則通知 LINE Pay 用戶付款未能成功完成。

使用 confirmUrl 的範例

使用 confirmUrl 的範例如下所示，其中商家所提供的 confirmUrl 為 "<http://testmall.com/pay/result>"。

請求 (LINE Pay -> 商家)

```
https://testmall.com/pay/result?orderId=2014xxx1232132&transactionId=201408011234567890
```

回應

```
// 若成功
HTTP/1.1 200 OK
Server:Apache-Coyote/1.1
Accept-Charset:utf-8
Content-Type:text/html
Content-Length:2
Date:Sat, 11 Oct 2014 02:45:48 GMT

// 若失敗
HTTP/1.1 404 NOT_FOUND
Server:Apache-Coyote/1.1
Accept-Charset:utf-8
Content-Type:text/html
Content-Length:4
Date:Sat, 11 Oct 2014 02:45:48 GMT
```

Sandbox 使用方法

在 LINE Pay 的 Sandbox 環境做測試時，需要使用記載在此連動集成 API Endpoint 明細上的 sandbox 來進行付款，付款畫面要由商家自行準備。在 Sandbox 環境試圖付款即將會如下去做處理。

※ 特殊事項

在 Sandbox 環境不管使用電腦或手機都需要登入 Channel Log-in。

(但，在實際環境上連動時，手機付款因為它已透過 LINE 收集使用者的 MID 資訊，不需登入 Channel Log-in。)

Sandbox PC 付款

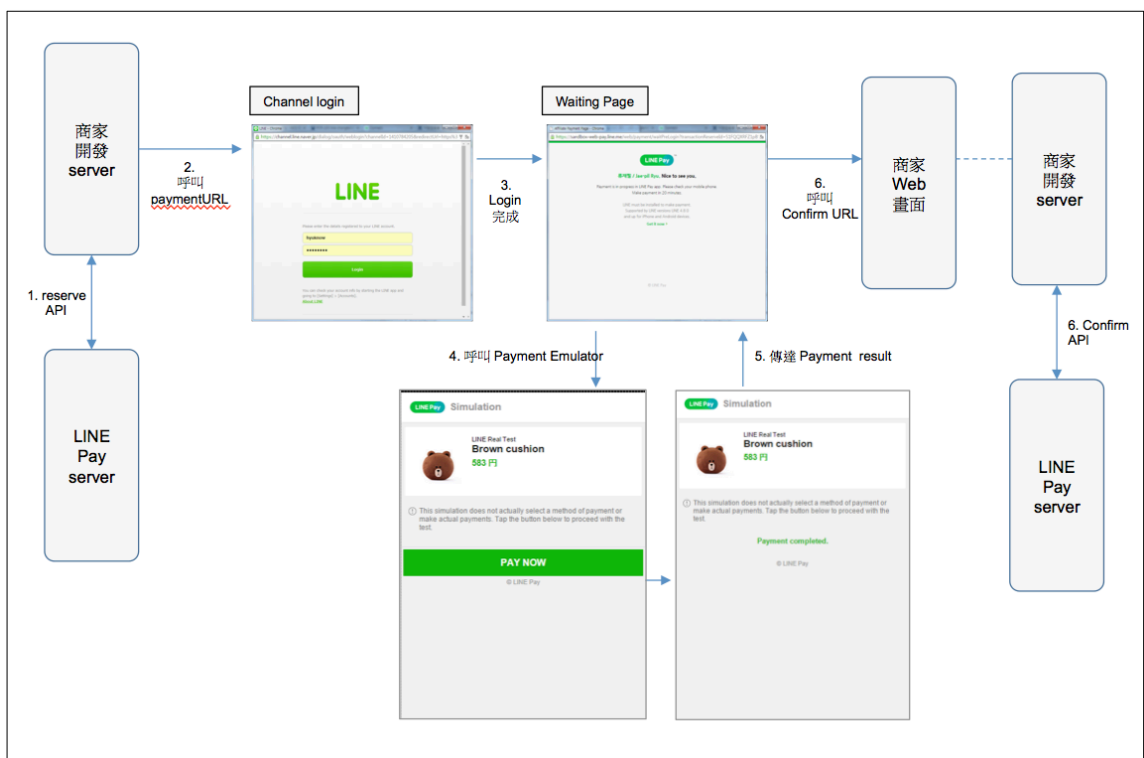


圖 10 Sandbox PC 進行付款畫面之例

Sandbox 手機付款

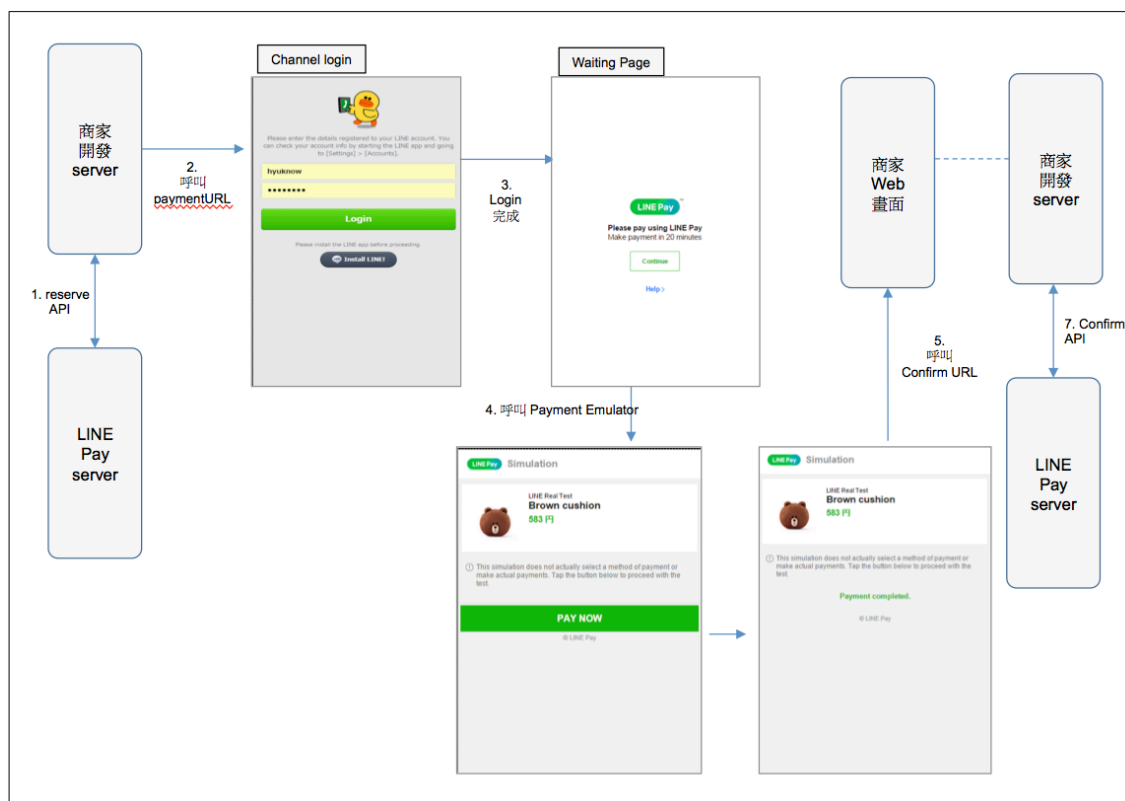


圖 11 Sandbox PC 進行付款畫面之例

Sandbox OneTimeKey 付款

商家可讀取測試人員在 LINE Pay app 適用 “My Code” 來顯示的二維碼或條碼來進行付款。

在 Sandbox 用 web page 形式去提供 “My Code”，可以測試付款。在 Sandbox 環境下，商家讀完 My Code 後再來呼叫 Reserve API，顯示 My Code 的該頁面會轉移成模擬付款頁面。

- Sandbox My code url : <https://sandbox-web-pay.line.me/web/sandbox/payment/otk>