

電子商店網站支付實務

1

修正問題

➡ [更新cart到最新版](#)

➡ <https://github.com/bmentges/django-cart>

➡ 修改 cart.py 及 models.py

➡ C:\Users\user\AppData\Local\Programs\Python\Python36-32\Lib\site-packages\cart

```
fields\file.py", line 134, in __init__
    super(FilerFileField, self).__init__(**kwargs)
TypeError: __init__() missing 1 required positional argument: 'on_delete'
```

➡ 修正 部份套件 沒有 'on_delete'

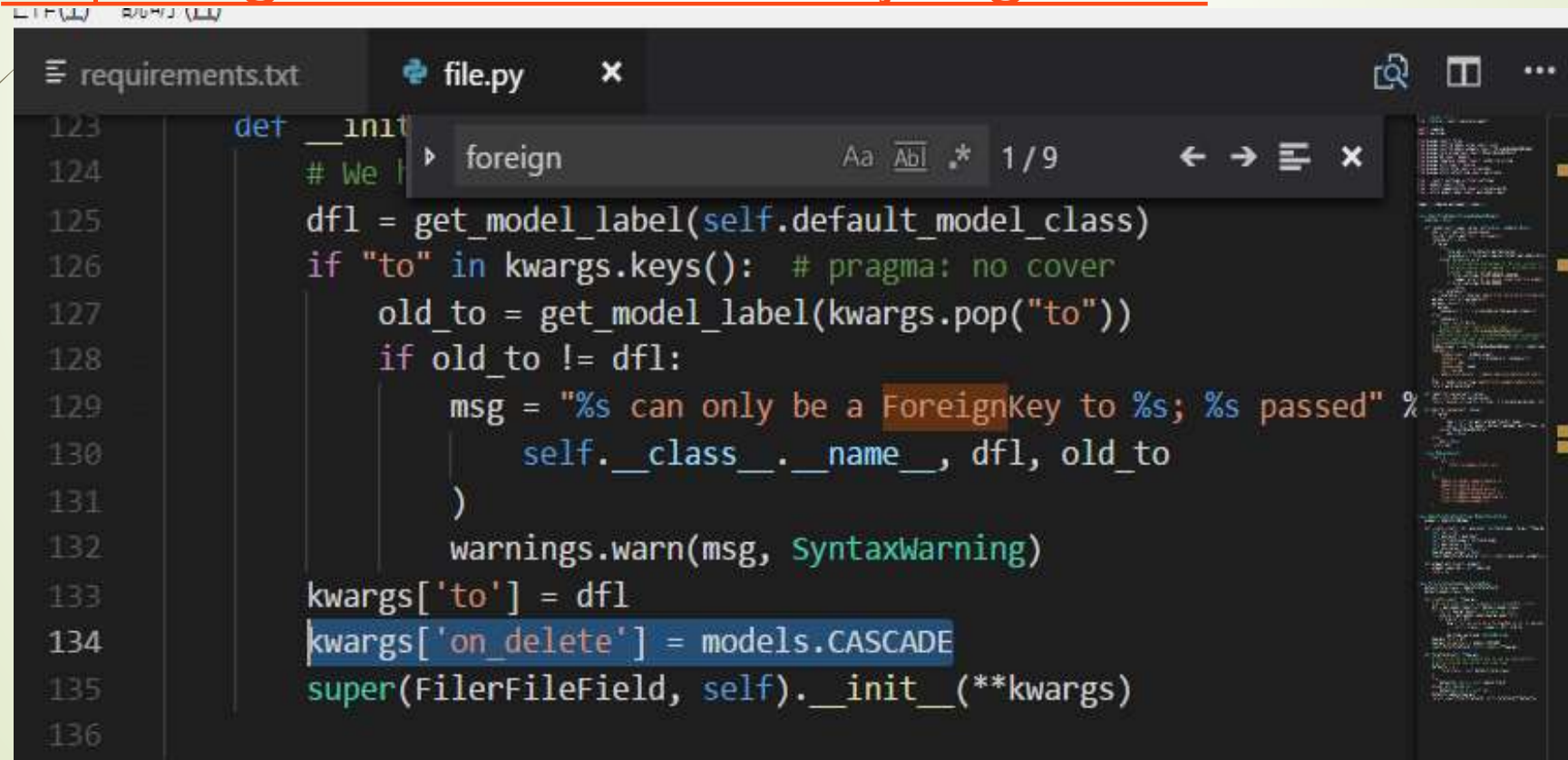
➡ **django報錯 : TypeError: __init__() missing 1 required positional argument: 'on_delete'**

➡ <https://www.itread01.com/content/1545016715.html>

➡ <https://www.cnblogs.com/imyalost/p/8506766.html>

file.py

- C:\Users\user\AppData\Local\Programs\Python\Python36-32\Lib\site-packages\filer\fields
- 新增第134行 : `kwargs['on_delete'] = models.CASCADE`
- <https://github.com/divio/django-filer>




```
requirements.txt  file.py  x
123 def __init__(self, *args, **kwargs):
124     # We are a ForeignKey
125     dfl = get_model_label(self.default_model_class)
126     if "to" in kwargs.keys(): # pragma: no cover
127         old_to = get_model_label(kwargs.pop("to"))
128         if old_to != dfl:
129             msg = "%s can only be a ForeignKey to %s; %s passed" % (
130                 self.__class__.__name__, dfl, old_to
131             )
132             warnings.warn(msg, SyntaxWarning)
133     kwargs['to'] = dfl
134     kwargs['on_delete'] = models.CASCADE
135     super(FilerFileField, self).__init__(**kwargs)
136
```

- 
- 編輯filer 於 settings.py 的路徑
 - MEDIA_URL = '/media/filer/'
 - MEDIA_ROOT =
'C:/Users/user/Downloads/mshop_modify2django2/media/filer'
 - STATIC_ROOT =
'C:/Users/user/Downloads/mshop_modify2django2/staticfiles'
 - FILER_STORAGES = {...}

 - PayPal 收款人帳號設定 settings.py
 - PAYPAL_TEST=True
 - PAYPAL_RECEIVER_EMAIL = '你的email'

 - 登入帳號：admin
 - 密碼：123456@abc



最後進行 migrate

- 安裝套件
- `pip install -r requirement.txt`
- `python manage.py makemigrations mysite`
- `python manage.py makemigrations cart`
- `python manage.py makemigrations`
- `python manage.py migrate mysite`
- `python manage.py migrate cart`
- `python manage.py migrate`


12.3

電子支付功能

在網站上直接讓會員可以線上付款要考慮的事項很多，主要是有許多網路安全上的考量，而本書的主要對象為初學者，為了避免讓內容太過於繁雜，我們簡化了許多的地方，因此，在學習了本章的內容之後，如果你有把自己的電子商店網站上線的需求的話，一定要再針對網站安全性的議題審慎評估過才行。

12.3.1 建立付款流程

在進入如何串接電子支付工具之前要先對於自己的網站之付款流程做好規劃。現在許多和部落格網站結合的電子支付按鈕工具，基本上都是在網路銀行上先建立好一個付款的

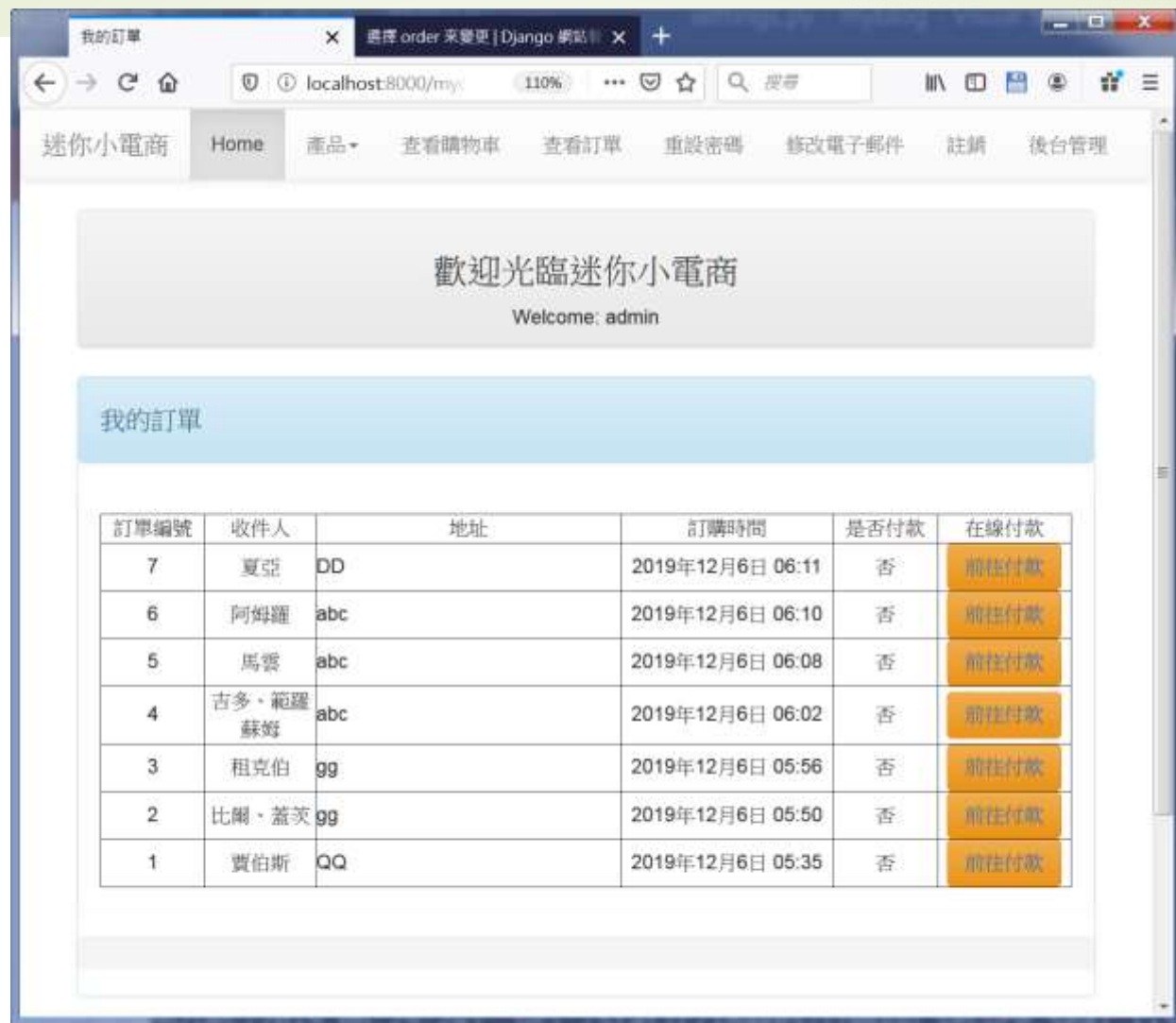


按鈕，然後把這個按鈕的代碼複製下來放在網站中顯示，在使用者按下此按鈕之後立即會前往該銀行的付款介面進行付款的操作。這樣的方法很簡單，但是卻沒有辦法做到和購物車結合的功能，每一個按鈕只有一個固定的付款金額，也因此如果有很多的品項，就必須為每一個品項建立個別的按鈕，對於同時販售許多商品的全功能型電子商店來說就非常不便。

本堂課所介紹的迷你小電商網站本身就具備有購物車的功能，此購物車的內容會隨著會員的操作而有所變化，當然每一次的金額也都不一定會一樣，也因此無法使用前段文字中所述的付款按鈕來做操作，而唯一的方式就是透過 API 的方式和付款網站進行串接，而這也是我們在本堂課所要教學的內容。

建立付款流程之規劃

1. 會員檢視所有的產品，在每一產品的下方均有加入購物車按鈕
2. 在選購了所有需要的商品之後，可以使用檢視購物車的連結查看目前在購物車內的產品內容以及數量，可以在此網頁中修正欲購買的內容。
3. 在檢視購物車的網頁中可以按下訂單按鈕，進入建立新訂單的網頁
4. 在建立新訂單的網頁中，會員需填入個人連絡資料才能夠正式把訂單放在資料庫中
5. 下完訂單之後會自動引導到檢視現有訂單的頁面，在每一筆未付款的訂單之後均有一個付款的按鈕。
6. 按下付款按鈕之後，再一次檢視詳細的訂購內容，然後使用線上付款的按鈕進行付款的操作。



和圖 12-19 不一樣的地方在於後面多加上了前往付款的按鈕，也就是在新增了一個叫做「線上付款」的欄位，然後判斷 `order.paid` 這個變數的內容，如果不是 `True` 的話則建立一個 `<button>` 的標籤，並建立到 `/payment` 網址的連結，並以 `order.id` 為此網址的參數。其主要的修正如下所示：

```
<div class='panel panel-body'>
  {% for order in orders %}
  {% if forloop.first %}
  <table border=1>
    <tr>
      <td width=100 align=center> 訂單編號 </td>
      <td width=100 align=center> 收件人 </td>
      <td width=300 align=center> 地址 </td>
      <td width=200 align=center> 訂購時間 </td>
      <td width=100 align=center> 是否付款 </td>
      <td width=100 align=center> 線上付款 </td>
    </tr>
  {% endif %}
  <div class='listgroup'>
    <div class='listgroup-item'>
      <tr>
        <td align=center>{{ order.id }}</td>
        <td align=center>{{ order.full_name }}
      </td>
```

```

</td>
"是, 否"}}</td>

/'> 前往付款 </a>

<td align=center>{{ order.address }}</td>
<td align=center>{{ order.created_at }}

<td align=center>{{ order.paid | yesno:
"% if not order.paid %"
<button class='btn btn-warning'>
  <a href='/payment/{{order.id}}
  </button>
  {% endif %}
</td>
</tr>
</div>
</div>
{% if forloop.last %}
</table>
{% endif %}
{% empty %}
  <em> 沒有處理中的訂單 </em>
{% endfor %}
</div>

```

所有未付款的訂單之後都會有一個「前往付款」的按鈕，按下該按鈕之後即可看到如圖 12-23 所示的訂單細項的摘要畫面，以及執行線上付款的按鈕，在此例為 PayPal。

選擇您的付款方式 | 選擇 order 來變更 | Django 網站

localhost:8000/pay 110%

迷你小電商 Home 產品 查看購物車 查看訂單 重設密碼 修改電子郵件 註銷 後台管理



歡迎光臨迷你小電商

Welcome: admin


在線付款 (訂單編號: 2)

產品名稱	單價	數量	小計
皮卡丘	30.00	1	30.00
GX-21	9999.00	1	9999.00

Buy Now

VISA    

NT\$:10029.00元

  本網站僅為教學用途，圖片均取自網路，如有侵權請告知，將立即撤下。
Copyright 2022 <http://qqggg.com>. All rights reserved.

由於在此時已知道訂單的編號以及付款的金額，因此此時的按鈕就等於是臨時訂製的付款按鈕，它可以把項目、訂單編號以及付款金額正確地傳遞給線上支付網站以完成付款的動作，而在付款完成之後，此筆訂單也會被註記為已付款 (paid=True)，方便商店管理員的核帳工作。

12.3.2 建立 PayPal 付款連結

PayPal 幾乎是通行全球的線上支付服務，在國內也有非常多的網路使用者擁有 PayPal 帳號可以進行線上付款，因此在電子商店網站上支援 Paypal 也是非常重要的功能。同樣地，PayPal 在 Django 中也有現成的模組可以使用，就叫做 django-paypal，同樣是以 pip 安裝即可，如下所示：

```
$ pip install django-paypal
```

接著在 settings.py 的 INSTALLED_APPS 區塊中加入 'paypal.standard.ipn'，同時也在 settings.py 中把 PAYPAL_TEST 這個常數先設定為 True，表示此付款機制要做為測試之用，等日後上線之後再把它設定為 False 即可，另外還有一個 PAYPAL_REVEIVER_EMAIL 常數則是設定為要收款的 PayPal 電子郵件帳號。設定完成之後，要使用 python manage.py migrate 做資料庫的同步動作。

urls.py 要加入以下的網址樣式，以便接收來自 PayPal 網站的付款程序完成通知：

```
url(r'^paypal/', include('paypal.standard.ipn.urls')),
```

除此之外，還要再加上另外 3 個網址，分別用來處理帳務以及顯示完成付款和取消付款動作的相關操作以及訊息，如下所示：

```
url(r'^payment/(\d+)/$', views.payment),  
url(r'^done/$', views.payment_done),  
url(r'^canceled/$', views.payment_canceled),
```

views.payment.done 沒有什麼特別之處，主要就是在會員完成 PayPal 之付款流程之後會回到的頁面，一般都是顯示感謝的訊息即可，如下所示：

```
@csrf_exempt  
def payment_done(request):  
    template = get_template('payment_done.html')  
    request_context = RequestContext(request)
```



```
request_context.push(locals())
html = template.render(request_context)
return HttpResponse(html)
```

由於此網址是由 PayPal 呼叫，會遭遇到 CSRF 的問題，因此我們需在 views.py 的開頭先引入以下的修飾元，並把 @csrf_exempt 加在 payment_done 函數之前：

```
from django.views.decorators.csrf import csrf_exempt
```

同樣地，payment_canceled 也是一樣的，如下所示：

```
@csrf_exempt
def payment_canceled(request):
    template = get_template('payment_canceled.html')
    request_context = RequestContext(request)
    request_context.push(locals())
    html = template.render(request_context)
    return HttpResponse(html)
```

至於 payment_done.html 則如下所示：

```
<!-- payment_done.html (mshop project) -->
{% extends "base.html" %}
{% block title %}Pay using PayPal{% endblock %}
{% block content %}
<div class='container'>
  {% for message in messages %}
    <div class='alert alert-{{message.tags}}'>{{ message }}</div>
  {% endfor %}
  <div class='row'>
    <div class='col-md-12'>
      <div class='panel panel-default'>
        <div class='panel-heading' align=center>
          <h3>歡迎光臨迷你小電商 </h3>
          {% if user.socialaccount_set.all.0.extra_data.name %}
            {{user.socialaccount_set.all.0.extra_data.
name}}<br/>
            <img src='{{user.socialaccount_set.all.0.get_
avatar_url}}' width='100'>
          {% else %}
            Welcome: {{ user.username }}
          {% endif %}
        </div>
      </div>
    </div>
  </div>
</div>
```



```
        </div>
    </div>
    <div class='row'>
        <div class='col-sm-12'>
            <div class='panel panel-info'>
                <div class='panel panel-heading'>
                    <h4>從 PayPal 付款成功 </h4>
                </div>
                <div class='panel panel-body'>
                    感謝您的支持，我們會儘速處理您的訂單。
                </div>
                <div class='panel panel-footer'>
                </div>
            </div>
        </div>
    </div>
</div>
{% endblock %}
```

而 payment_canceled.html 基本上內容和 payment_done.html 是一樣的，就只有呈現的訊息不太相同，如下所示：

... 略 ...

```
<div class='row'>
  <div class='col-sm-12'>
    <div class='panel panel-info'>
      <div class='panel panel-heading'>
        <h4>您剛剛取消了 PayPal 的付款 </h4>
      </div>
      <div class='panel panel-body'>
        <p>請再次檢查您的付款，或是返回 <a href='/myorders/'>我的訂單
</a> 選用其他付款方式。 </p>
      </div>
      <div class='panel panel-footer'>
      </div>
    </div>
  </div>
</div>
```

... 略 ...

真正的重點為如何在網站上完成利用 PayPal 付款的工作，此工作主要分成兩大部份，第一部份是建立含有正確資料的付款按鈕，而第二個部份則是在會員完成付款動作之後也會跟著更新資料庫中該筆訂單的付款狀態。先來看第一部份，此部份的主要內容寫在 views.payment 中，如下所示：

```
@login_required
def payment(request, order_id):
    all_categories = models.Category.objects.all()
    try:
        order = models.Order.objects.get(id=order_id)
    except:
        messages.add_message(request, messages.WARNING, " 訂單編號錯誤，無法處理
        付款。")
        return redirect('/myorders/')
    all_order_items = models.OrderItem.objects.filter(order=order)
    items = list()
    total = 0
    for order_item in all_order_items:
        t = dict()
        t['name'] = order_item.product.name
        t['price'] = order_item.product.price
        t['quantity'] = order_item.quantity
        t['subtotal'] = order_item.product.price * order_item.quantity
        total = total + order_item.product.price
    items.append(t)
```



```
host = request.get_host()
paypal_dict = {
    "business": settings.PAYPAL_REVEIVER_EMAIL,
    "amount": total,
    "item_name": "迷你小電商貨品編號:{}".format(order_id),
    "invoice": "invoice-{}".format(order_id),
    "currency_code": 'TWD',
    "notify_url": "http://{}{}".format(host, reverse('paypal-ipn')),
    "return_url": "http://{}done/".format(host),
    "cancel_return": "http://{}canceled/".format(host),
}
paypal_form = PayPalPaymentsForm(initial=paypal_dict)
template = get_template('payment.html')
request_context = RequestContext(request)
request_context.push(locals())
html = template.render(request_context)
return HttpResponse(html)
```


此段程式碼分成 2 個部份，前半段的目的是在於透過輸入的訂單編號找出訂單的細項，然後使用一個迴圈把它放在 items 這個串列變數，同時也順便計算出總金額放在 total 中。後半段的目的是在於使用已知的資訊建立起相對應的 PayPal 按鈕。建立 PayPal 按鈕的方法很簡單，就是填寫 paypal_dict 這個字典的內容，然後使用此字典建立 paypal_form 表單，而要建立此表單，則是使用 django-paypal 提供的 PayPalPaymentsForm 這個類別，也因此 views.py 的前面也要引入這個類別才行，另外因為有用到 settings.py 中的常數，所以也要引入 settings，如下所示：

```
from django.conf import settings
from paypal.standard.forms import PayPalPaymentsForm
from django.core.urlresolvers import reverse
```

至於 paypal_dict 字典的內容，說明如下表：

paypal_dict的字典內容說明

key	說明	程式中的內容
business	收款人的電子郵件，也就是站長在PayPal的收款帳號，設定在settings.py中的常數	settings.PAYPAL_RECEIVER_EMAIL
amount	收款金額	total
item_name	此品項名稱	"迷你小電商貨品編號:{}".format(order_id)
invoice	發票編號，中間的「-」不要漏掉，那是要用來分隔訂單編號用的符號	"invoice-{}".format(order_id)
currency_code	貨幣編碼，ISO-4217標準	'TWD'
notify_url	付款狀態通知用網址，使用reverse取得django-paypal設定的網址	"http://{}".format(host, reverse('paypal-ipn'))
return_url	程序完畢返回網址	"http://{/done/".format(host)
cancel_return	取消付款返回網址	"http://{/canceled/".format(host)

key	說明	程式中的內容
business	收款人的電子郵件，也就是站長 在 PayPal 的收款帳號，設定在 settings.py 中的常數	settings.PAYPAL_RECEIVER_EMAIL
amount	收款金額	total
item_name	此品項名稱	"迷你小電商貨品編號:{}".format(order_id)
invoice	發票編號，中間的「-」不要漏掉， 那是要用來分隔訂單編號用的符號	"invoice-{}".format(order_id)
currency_code	貨幣編碼，ISO-4217 標準	'TWD'
notify_url	付款狀態通知用網址，使用 reverse 取得 django-paypal 設定 的網址	"http://{}".format(host, reverse('paypal-ipn'))
return_url	程序完畢返回網址	"http://{/done}/".format(host)
cancel_return	取消付款返回網址	"http://{/canceled}/".format(host)

透過 `paypal_form = PayPalPaymentsForm(initial=paypal_dict)` 產生出一個實例 `paypal_form`，此變數被送到 `payment.html` 中渲染之後，即會產生正確的付款按鈕。
`payment.html` 的內容如下：

```
<!-- payment.html (mshop project) -->
{% extends "base.html" %}
{% block title %} 選擇您的付款方式 {% endblock %}
{% block content %}
<div class='container'>
{% for message in messages %}
    <div class='alert alert-{{message.tags}}'>{{ message }}</div>
```



```
{% endfor %}
  <div class='row'>
    <div class='col-md-12'>
      <div class='panel panel-default'>
        <div class='panel-heading' align=center>
          <h3>歡迎光臨迷你小電商 </h3>
          {% if user.socialaccount_set.all.0.extra_data.name %}
            {{user.socialaccount_set.all.0.extra_data.
name}}<br/>
            <img src='{{user.socialaccount_set.all.0.get_
avatar_url}}' width='100'>
          {% else %}
            Welcome: {{ user.username }}
          {% endif %}
        </div>
      </div>
    </div>
  </div>
  <div class='row'>
    <div class='col-sm-12'>
      <div class='panel panel-info'>
        <div class='panel panel-heading'>
          <h4>線上付款 (訂單編號: {{order.id}}) </h4>
```



```
</div>
<div class='panel panel-body'>
  {% for item in items %}
  {% if forloop.first %}
  <table border=1>
    <tr>
      <td width=300 align=center>產品名稱 </td>
      <td width=100 align=center>單價 </td>
      <td width=100 align=center>數量 </td>
      <td width=100 align=center>小計 </td>
    </tr>
  {% endif %}
  <div class='listgroup'>
    <div class='listgroup-item'>
      <tr>
        <td>{{ item.name }}</td>
        <td align=right>{{ item.price }}</td>
        <td align=center>{{ item.quantity }}</td>
        <td align=right>{{ item.subtotal }}</td>
      </tr>
```

```
        </div>
    </div>
    {% if forloop.last %}
</table>
    {% endif %}
    {% empty %}
        <em> 此訂單是空的 </em>
    {% endfor %}

    {{ paypal_form.render }}
</div>
<div class='panel panel-footer'>
    NT$:{{ total }}元
</div>
</div>
</div>
</div>
</div>
{% endblock %}
```

此檔案中最重要內容即為此行：{{ paypal_form.render }}，透過 render 會產生出正確金額的 PayPal 付款按鈕，如此即可完成如圖 12-23 所示的介面。此時，只要會員點擊此按鈕，過一小段時間之後，即會被引導至 PayPal 官網中進行付款的動作，如圖 12-24 所示。

Pay with a PayPal account

https://www.sandbox.paypal.com/cgi-bin/webscr?cmd=_flow&SESSION=sQ1uqj1wx4ETqg4CsN5j

skynet.tw@gmail.com


Your order summary

Descriptions	Amount
迷你小電燈產品編號:43 Item price: NT\$200 Quantity: 1	NT\$200
Item total	NT\$200
	Total NT\$200 TWD

Choose a way to pay

Pay with my PayPal account

Log in to your account to complete the purchase



Email

PayPal password

This is a private computer. [What's this?](#)

[Forgot email or password?](#)

Create a PayPal account

And pay with your debit or credit card

[Cancel and return to skynet.tw@gmail.com](#)

Site Feedback
PayPal. The safer, easier way to pay. For more information, read our [User Agreement](#) and [Privacy Policy](#).

Test Site

不過到目前為止我們只做了前半部，亦即建立正確的付款按鈕作業，在會員完成付款之後，我們還要再處理有訂單的資料內容，也就是把相對應的這張訂單註明為已付款，這點非常重要，請看下一小節的說明。

12.3.3 接收 PayPal 付款完成通知

PayPal 在處理完線上付款流程之後會另外再送一個 HTTP 資料給我們的網站，這也是為什麼我們在 `paypal_dict` 中要指定一個 `notify_url` 的原因。django-paypal 使用 Django 的 signal 機制來傳送這個通知給我們，因此，我們就要在一開始的時候指定一個處理這個訊號的函數，當發現 django-paypal 轉發此 signal 的時候可以立即處理，也就是判斷該付款是否已完成，如果沒問題的話，就找出指定編號的訂單記錄，然後把該訂單的 `paid` 欄位設定為 `True` 再加以儲存。為了確保我們設定的監聽用函數可以保證被系統載入且能保持運作中，我們把這個處理函數另外儲存為一個名為 `signal.py` 的檔案，也是放在 `mysite` 資料夾中和 `views.py` 同一個位置處，其內容如下：

```
from mysite import models
from paypal.standard.models import ST_PP_COMPLETED
from paypal.standard.ipn.signals import valid_ipn_received

def payment_notification(sender, **kwargs):
    ipn_obj = sender
    if ipn_obj.payment_status == ST_PP_COMPLETED:
        order_id = ipn_obj.invoice.split('-')[-1]
        order = models.Order.objects.get(id = order_id)
        order.paid = True
        order.save()

valid_ipn_received.connect(payment_notification)
```

django-payapl使用的信號叫做 `valid_ipn_received`，因此除了定義要處理的函數名為 `payment_notification` 之外，也要在本檔案的最後一行以 `valid_ipn_received.connect(payment_notification)` 的方式把此函數註冊到接收此信號的處理函數中。當 PayPal 傳遞訊息過來時，此函數即會被呼叫啟用，並可以透過 `ipn_obj=sender` 的方式取得訊息的物件，藉由此物件的屬性得知其中的內容，所有的內容在 PayPal 的開發者網站 (<https://developer.paypal.com/webapps/developer/docs/classic/ipn/integration-guide/IPNandPDTVariables/>) 中有詳細的說明，理論上為了安全起見我們應該要檢查更多的項

我的訂單 Summary - PayPal 選擇 PayPal

localhost:8000/admin/ipn/paypalipn

Django 管理

首頁 > Ipn > PayPal IPNs

✔️ PayPal IPN "PayPalIPN: 1" 已成功變更。

選擇 PayPal IPN 來變更

搜尋

< 2019 12月5日

動作: 1 中 0 個被選

UNICODE	FLAG	FLAG INFO	INVOICE	CUSTOM	PAYMENT STA
<IPN: Transaction 3afg4-86s4fg- s6f54sd6f>	✔️	2			ST_PP_COM

1 PayPal IPN

我的訂單 Summary - PayPal 選擇 order 來變更 | Django 網站

localhost:8000/admin/mysite/order/

Django 管理

首頁 > Mysite > Orders

選擇 order 來變更

動作: 7 中 0 個被選

- ORDER
- Order:7
- Order:6
- Order:5
- Order:4
- Order:3
- Order:2**
- Order:1

7 orders

全部
aa

目看看是否符合。但是為了簡化示範的內容，我們僅僅檢查其 `payment_status` 是否為 `ST_PP_COMPLETED`，如果是的話表示在 PayPal 的付款動作已完成無誤，如果是這樣的話，我們在程式中就先從 `invoice` 中拆解出訂單編號，然後根據此編號把該筆訂單找出來，利用 `order.paid=True` 設定為已付款，然後再儲存該筆記錄，就算是大功告成了。

那麼，如何確保此函數在網站一開始的時候就能夠載入呢？我們用了以下的技巧，首先，在同一個資料夾之下再建立一個 `apps.py` 的檔案，內容如下所示：

```
from django.apps import AppConfig
class PaymentConfig(AppConfig):
    name = 'mysite'
    verbose_name = 'Mysite'

    def ready(self):
        import mysite.signal
```

Payment status :

ST_PP_COMPLETED

`ready` 這個函數可以用來做應用程式初始化工作，一般來說都會在這個函數中註冊 `signal`，因此，我們在這個函數中把剛剛定義的信號處理函數做載入的動作，此外，再於 `mysite` 中的 `__init__.py` 中加入下面這行指令：

```
default_app_config = 'mysite.apps.PaymentConfig'
```



確保我們在應用程式初始化載入 `mysite` 的時候，可以把我們自訂的應用程式環境設定也能夠載入做自訂的工作，如此再重新載入整個應用程式時，就可以正確地收到並處理來自於 PayPal 的通知了。

12.3.4 測試 PayPal 付款功能

經過以上的設定，我們的網站已經可以正確地接收訂單並使用 PayPal 做收款的動作了，但是，要如何測試呢？總不能直接拿別人的帳號來做交易吧？！別擔心，在 PayPal 的開發者網站（<https://developer.paypal.com/>）中可以設定測試專用的帳號，當然，你必須要先有 PayPal 的一般帳號才行。前往該網站之後，使用原本的 PayPal 帳號登入之後，即會進入如圖 12-25 所示的畫面。





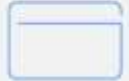


PayPal Developer Document x 選擇 order 來變更 | Django 網站 x +

https://developer.paypal.com

PayPal Developer Docs APIs Support Search <https://developer.paypal.com/> Log into Dashboard

Explore our payment products and solutions

<h3>Checkout</h3>  <p>Help increase sales with Smart Payment Buttons.</p>	<h3>PayPal Commerce Platform</h3> <p>Get our global, end-to-end payment solutions and technologies for marketplaces, e-commerce solution providers, and more.</p>
<h3>Subscriptions</h3>  <p>Get paid consistently with a subscription plan.</p>	<h3>Payouts</h3>  <p>Send money to many PayPal or Venmo recipients at the same time.</p>
<h3>Invoicing</h3>  <p>Get paid fast when you send professional, customizable invoices from</p>	<h3>Accept card payments</h3>  <p>Get paid by credit card on the web, in your app, or in person. See our</p>

FEEDBACK

DASHBOARD

My Apps & Credentials

My Account

SANDBOX

Accounts

Notifications

API Calls

IPN Simulator

Webhooks Events

MOCK

Webhooks Simulator

Credit Card Generator

Negative Testing

My Apps & Credentials

Sandbox

Live

REST API apps

Get started quickly by using the Default Application credentials for testing PayPal REST APIs on the Sandbox environment.

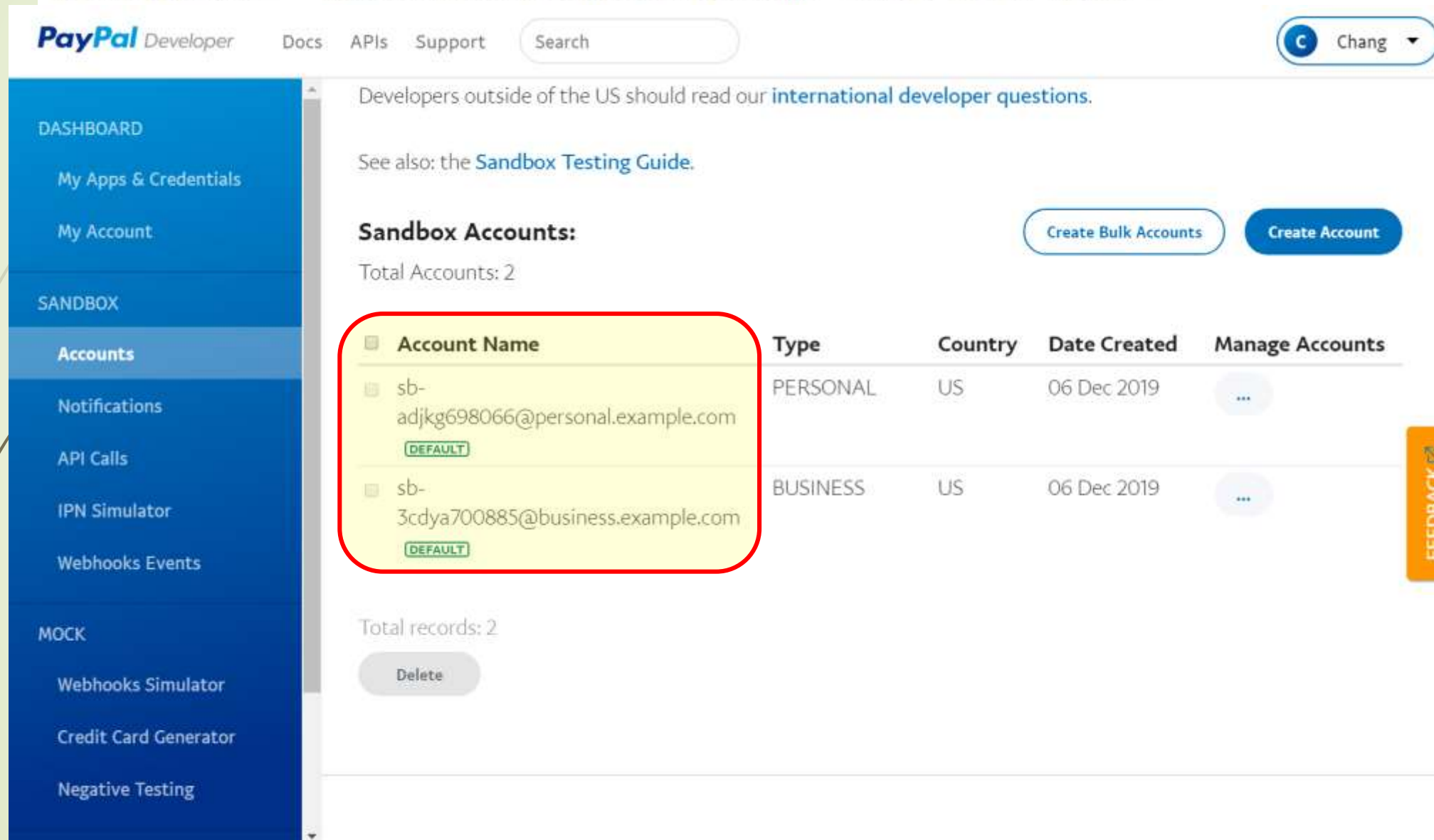
App Name	Type	Actions
Default Application	REST	System generated, no actions available.

Create App

NVP/SOAP API apps

To access these apps, use the Live toggle to leave the testing environment.

在左側的 Sandbox 標籤組中有一個 Accounts 的連結，即為申請測試用帳號的地方。在申請之後，一般都可以有 2 個帳號可供測試，如圖 12-27 所示。



PayPal Developer Docs APIs Support Search C Chang

Developers outside of the US should read our [international developer questions](#).

See also: the [Sandbox Testing Guide](#).

Sandbox Accounts: Create Bulk Accounts Create Account

Total Accounts: 2

Account Name	Type	Country	Date Created	Manage Accounts
sb-adjkg698066@personal.example.com <small>DEFAULT</small>	PERSONAL	US	06 Dec 2019	...
sb-3cdya700885@business.example.com <small>DEFAULT</small>	BUSINESS	US	06 Dec 2019	...

Total records: 2 Delete

FEEDBACK

DASHBOARD

My Apps & Credentials

My Account

SANDBOX

Accounts

Notifications

API Calls

IPN Simulator

Webhooks Events

MOCK

Webhooks Simulator

Credit Card Generator

Negative Testing

Developers outside of the US should read our [international developer questions](#).

See also: the [Sandbox Testing Guide](#).

Sandbox Accounts:

Create Bulk Accounts

Create Account

Total Accounts: 2

Account Name	Type	Country	Date Created	Manage Accounts
sb-adjkg698066@personal.example.com <small>DEFAULT</small>	PERSONAL	US	06 Dec 2019	...
sb-3cdya700885@business.example.com <small>DEFAULT</small>	BUSINESS	US	06 Dec 2019	...

Total records: 2

Delete

- View/Edit Account
- Duplicate Account
- View Notifications

FEEDBACK

DASHBOARD

My Apps & Credentials

My Account

SANDBOX

Accounts

Notifications

API Calls

IPN Simulator

Webhooks Events

MOCK

Webhooks Simulator

Credit Card Generator

Negative Testing

Account Details



Profile

API Credentials

Funding

Settings

First Name:

John

Last Name:

Doe

Email ID:

sb-3cdya700885@business.example.com

System Generated Password:

FJ33oxT&

Password:

[Change password](#)

Note: If your system generated password is changed, your new password will not be displayed because of one-way password encryption.

Phone Number:

4085934812

Accounts [Create Account](#)

Manage Accounts

Account Details

Profile API Credentials Funding Settings

Payment Review

Review sandbox payments before they are completed.

 On Off

Negative Testing

Recreate error conditions to test error handling.

 On Off

PayPal Credit

Enable/Disable PayPal EZ Payments for your sandbox account.

 On Off



NT\$10,029 TWD

Pay with PayPal

English

sb-3cdya700885@business.example.com

Password

Log In

Having trouble logging in?

or

Create an Account



New. Faster. Easier.

Welcome to the new PayPal checkout!
The security you rely on – now even faster.
It's everything checkout should be.



NT\$10,029 TWD

Hi, John!

Choose a way to pay

- Balance
- VISA Visa x-9082
- CREDIT UNION 1 x-0870
- [+ Add a debit or credit card](#)

Continue



PayPal is the safer, easier way to pay

No matter where you shop, we keep your financial information secure.

[Cancel & Return to Merchant](#)

[Policies](#) [Terms](#) [Privacy](#) [Feedback](#) © 1999 - 2019



NT\$10,029 TWD

Hi, John!

Pay with

[Change >](#)



Balance

\$338.27
USD

Make my balance my preferred way to pay

PayPal rate: 1 USD = 29.6478 TWD

View [PayPal Policies](#) and your payment method rights.

Pay Now



PayPal is the safer, easier way to pay

No matter where you shop, we keep your financial information secure.

PayPal Checkout - Review your order | 選擇 order 來變更 | Django 網站

https://www.sandbox.paypal.com/webapps/hermes

NT\$10,029 TWD

Hi, John!

Pay with

Balance

Change 1

\$338.27

Make my balance my preferred way to pay

PayPal rate: 1 USD = 29 TWD

Processing...

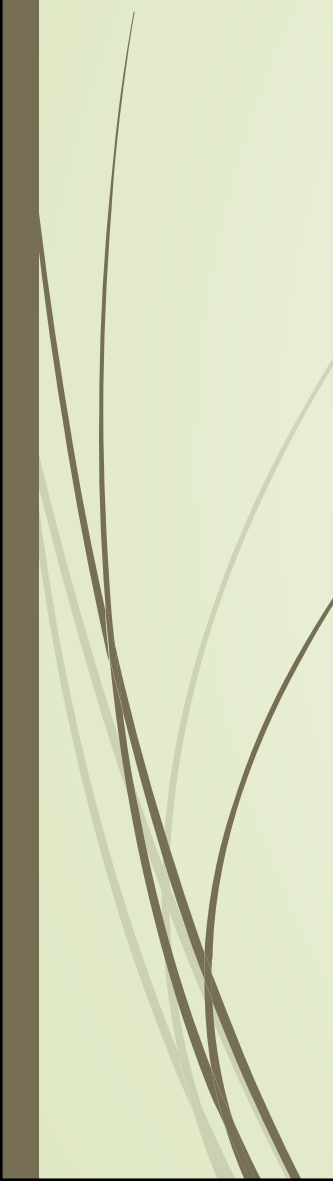
Pay Now

PayPal is the safer, easier way to pay

No matter where you shop, we keep your financial information secure.


Cancel & Return to Merchant


Policies Terms Privacy Feedback © 1999 - 2019



PayPal Checkout - Payment c X 選擇 order 來變更 | Django 網站 X +

https://www.sandbox.paypal.com/webapps/hermes





You paid NT\$10,029 TWD

to the merchant
[Details](#)

Get PayPal Notifications in Messenger

Paid with

Balance	\$338.27 USD
---------	--------------


Purchase details

Receipt number: 1AT37588KN211135L

We'll send confirmation to:
sb-3cdya700885@business.example.com

PayPal Checkout - Payment c X 選擇 order 來變更 | Django 網站 X +

https://www.sandbox.paypal.com/webapps/hermes 搜尋



You paid NT\$10,029 TWD
to the merchant
[Details](#)

Get PayPal Notifications in Messenger

Paid with
Balance \$338.27 USD

Purchase details
Receipt number: 1AT37588KN211135L
We'll send confirmation to:
sb-3cdya700885@business.example.com

Return to Merchant

Policies Terms Privacy Feedback © 1999 - 2019

Pay using PayPal

選擇 order 來變更 | Django 網站

localhost:8000/done/ 110%

迷你小電商 Home 產品 查看購物車 查看訂單 重設密碼 修改電子郵件 註銷 後台管理

歡迎光臨迷你小電商

Welcome: admin

從PayPal付款成功

感謝您的支持，我們會盡快處理您的訂單。



本網站僅為教學用途，圖片均取自網路，如有侵權請告知，將立即撤下。
Copyright 2022 <http://qqggg.com>. All rights reserved.

此時我們再回到檢視訂單的地方，就可以看到訂單已被設定為已付款狀態了，如圖

歡迎光臨迷你小電商
Welcome: admin

我的訂單

訂單編號	收件人	地址	訂購時間	是否付款	在線付款
7	夏亞	DD	2019年12月6日 06:11	否	前往付款
6	阿姆羅	abc	2019年12月6日 06:10	否	前往付款
5	馬雲	abc	2019年12月6日 06:08	否	前往付款
4	吉多、範羅蘇姆	abc	2019年12月6日 06:02	否	前往付款
3	租克伯	gg	2019年12月6日 05:56	否	前往付款
2	比爾、蓋茨	gg	2019年12月6日 05:50	是	
1	賈伯斯	QQ	2019年12月6日 05:35	否	前往付款

那麼想知道此張訂單在客戶端中看起來的樣子嗎？沒問題，只要前往 Sandbox 頁面（請留意網址是 <https://www.sandbox.paypal.com>），如圖 12-34 所示。

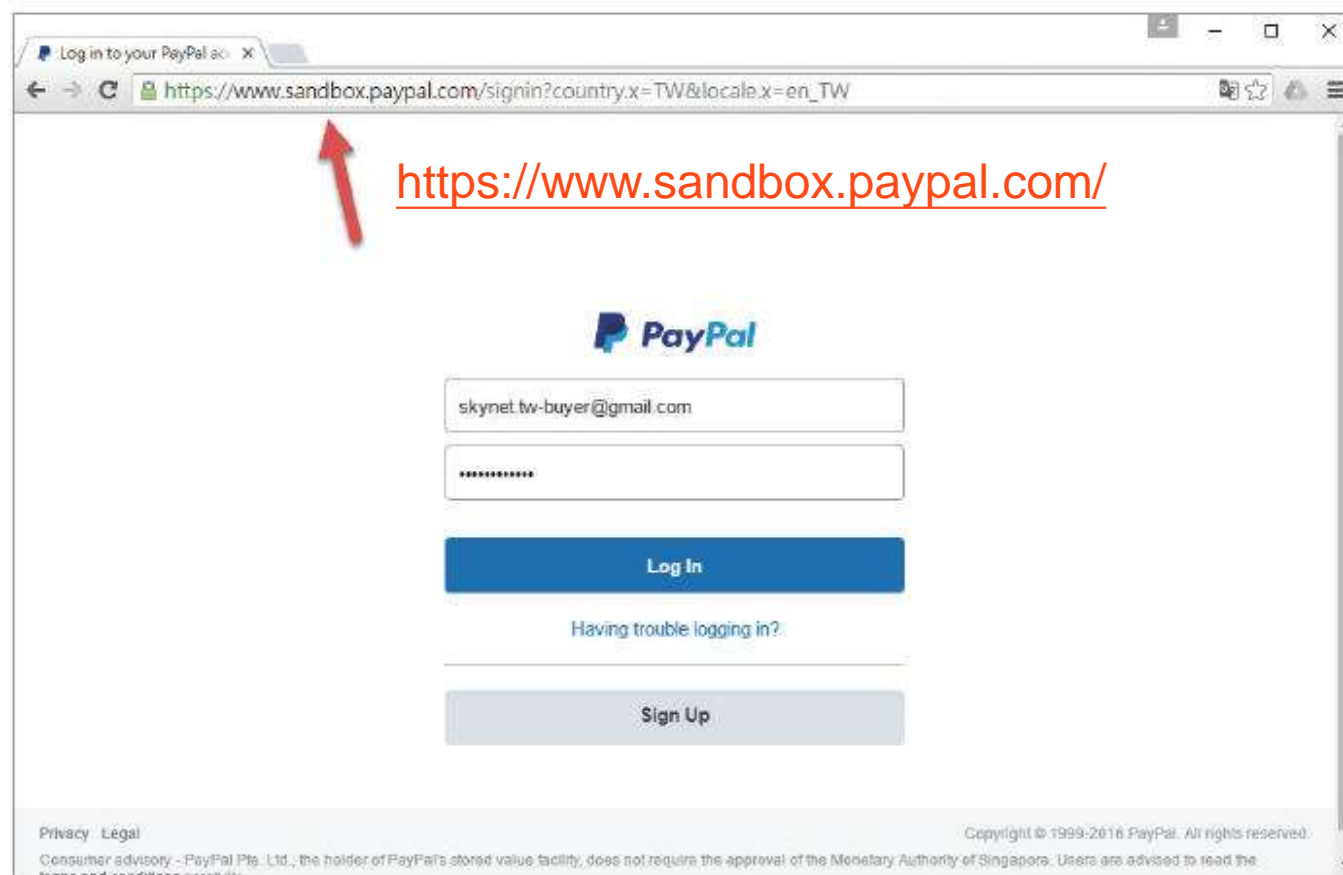
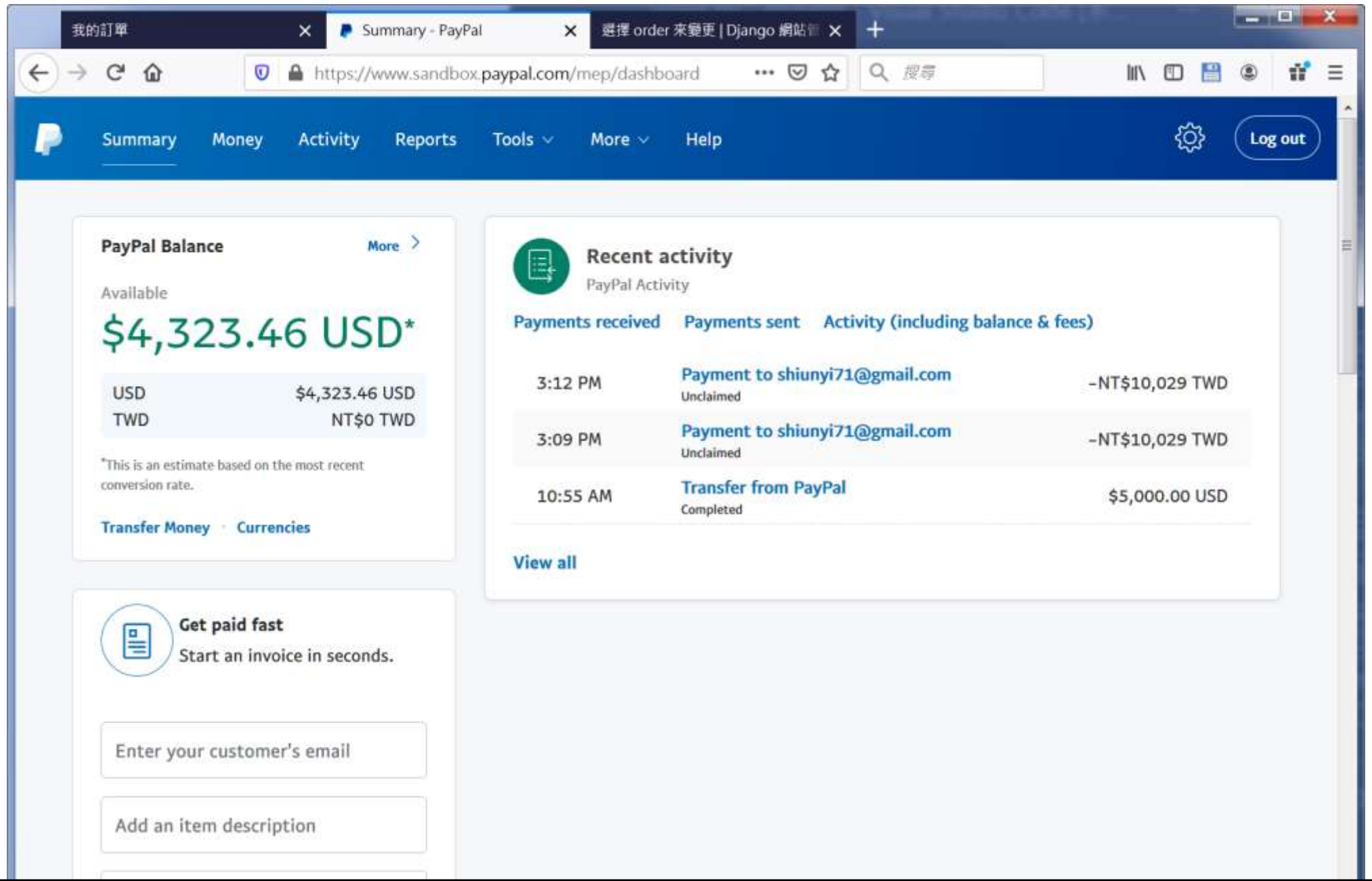


圖 12-34：PayPal 的 Sandbox 帳號專用網站

使用剛剛的測試帳號和密碼登入，就可看到所有測試訂單的樣子，如圖 12-35 所示



The screenshot shows the PayPal dashboard interface. At the top, there are browser tabs for '我的訂單', 'Summary - PayPal', and '選擇 order 來變更 | Django 網站'. The address bar shows 'https://www.sandbox.paypal.com/mep/dashboard'. The main navigation bar includes 'Summary', 'Money', 'Activity', 'Reports', 'Tools', 'More', and 'Help', along with a 'Log out' button. The dashboard is divided into several sections:

- PayPal Balance:** Shows an available balance of \$4,323.46 USD. Below this, a table shows the conversion: USD \$4,323.46 USD and TWD NT\$0 TWD. A note states: '*This is an estimate based on the most recent conversion rate.' Links for 'Transfer Money' and 'Currencies' are provided.
- Recent activity:** Titled 'PayPal Activity', it lists recent transactions:

Time	Description	Amount
3:12 PM	Payment to shiunyi71@gmail.com Unclaimed	-NT\$10,029 TWD
3:09 PM	Payment to shiunyi71@gmail.com Unclaimed	-NT\$10,029 TWD
10:55 AM	Transfer from PayPal Completed	\$5,000.00 USD

A 'View all' link is located below the activity list.
- Get paid fast:** Promotes starting an invoice in seconds. It includes input fields for 'Enter your customer's email' and 'Add an item description'.

習題

1. 請把分類檢視的選單改為使用側邊欄的方式來完成。
2. 請為範例網站中的詳細產品資訊加上MarkDown標記的功能。
3. 請試著把價錢的小數點移除。
4. 在12.2.3小節中如何避免被使用錯誤的產品編號執行到加入購物車的功能？
5. 在12.2.3小節的網站中只要把某一產品加入購物車就直接回首頁，請問這樣會造成什麼樣的困擾？如何解決？