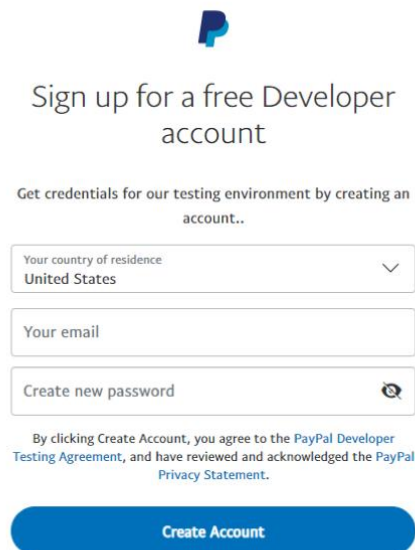


Setup Development Environment

1. Tạo tài khoản PayPal Developer

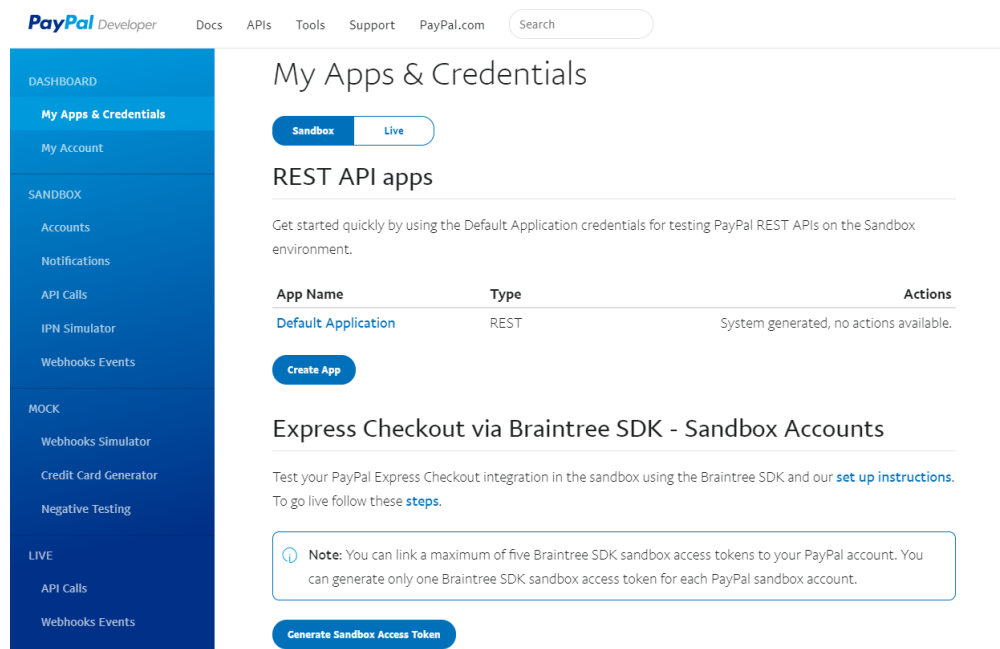
Đăng ký tài khoản PayPal Developer tại:

<https://www.paypal.com/signin/client?flow=provisionUser>



The image shows the PayPal Developer sign-up page. At the top is the PayPal logo. Below it, the text reads "Sign up for a free Developer account". Underneath, it says "Get credentials for our testing environment by creating an account..". There are three input fields: "Your country of residence" (with "United States" selected), "Your email", and "Create new password" (with a strength indicator). Below the fields, a small text block states: "By clicking Create Account, you agree to the PayPal Developer Testing Agreement, and have reviewed and acknowledged the PayPal Privacy Statement." At the bottom is a large blue "Create Account" button.

Đăng nhập vào dashboard



The image shows the PayPal Developer dashboard. The top navigation bar includes "PayPal Developer", "Docs", "APIs", "Tools", "Support", "PayPal.com", and a search bar. The left sidebar has a menu with categories: DASHBOARD (My Apps & Credentials, My Account), SANDBOX (Accounts, Notifications, API Calls, IPN Simulator, Webhooks Events), MOCK (Webhooks Simulator, Credit Card Generator, Negative Testing), and LIVE (API Calls, Webhooks Events). The main content area is titled "My Apps & Credentials" and has tabs for "Sandbox" and "Live". Below the tabs, it says "REST API apps" and "Get started quickly by using the Default Application credentials for testing PayPal REST APIs on the Sandbox environment." There is a table with one row: "Default Application" (App Name), "REST" (Type), and "System generated, no actions available." (Actions). Below the table is a "Create App" button. Further down, there is a section titled "Express Checkout via Braintree SDK - Sandbox Accounts" with instructions to test integration and follow steps to go live. A note states: "Note: You can link a maximum of five Braintree SDK sandbox access tokens to your PayPal account. You can generate only one Braintree SDK sandbox access token for each PayPal sandbox account." At the bottom is a "Generate Sandbox Access Token" button.

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

2. Tạo REST API app

Để gọi được các API của PayPal thì ta phải tạo một ứng dụng REST API (REST API apps), sau đó lấy các thông tin xác thực (credentials).

Mặc định PayPal đã tạo sẵn một REST API app với tên Default Application.

Để tạo ứng dụng REST API:

1. Chọn **Create App**
2. Nhập **App Name**
3. Nhấn nút **Create App**

PayPal Developer Docs APIs Tools Support PayPal.com Search

DASHBOARD

My Apps & Credentials

My Account

SANDBOX

Accounts

Notifications

API Calls

IPN Simulator

Webhooks Events

MOCK

Webhooks Simulator

Credit Card Generator

Negative Testing

LIVE

API Calls

Webhooks Events

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

Express Checkout via Braintree SDK - Sandbox Accounts

Test your PayPal Express Checkout integration in the sandbox using the Braintree SDK and our [set up instructions](#). To go live follow these [steps](#).

Note: You can link a maximum of five Braintree SDK sandbox access tokens to your PayPal account. You can generate only one Braintree SDK sandbox access token for each PayPal sandbox account.

Generate Sandbox Access Token

PayPal Developer Docs APIs Tools Support PayPal.com Search

DASHBOARD

My Apps & Credentials

My Account

SANDBOX

Accounts

Notifications

API Calls

IPN Simulator

Webhooks Events

MOCK

Webhooks Simulator

Credit Card Generator

Negative Testing

Create New App

Before you create your new app, let us know what kind of solution you're looking for.

Application Details

App Name

My App

Sandbox Business Account

sb-pclks1988345@business.example.com

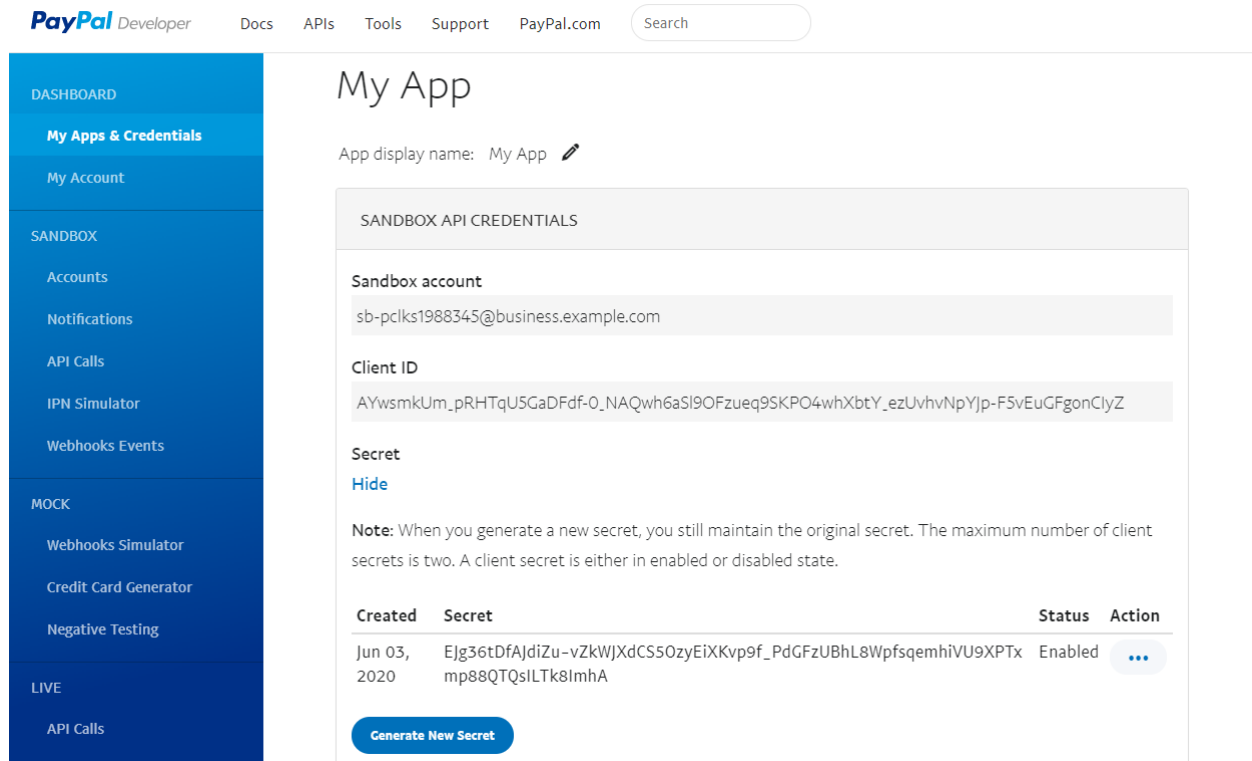
Create App

3. Lấy thông tin xác thực API (API credentials)

Thông tin xác thực API bao gồm: **Client ID** và **Secret**.

Hai thông tin này bắt buộc phải có khi gọi API của PayPal để giao dịch.

Lưu hai thông tin này lại để sau này sử dụng để gọi API của PayPal.



The screenshot shows the PayPal Developer dashboard for an application named 'My App'. The left sidebar contains navigation links for Dashboard, My Apps & Credentials, My Account, Sandbox (Accounts, Notifications, API Calls, IPN Simulator, Webhooks Events), Mock (Webhooks Simulator, Credit Card Generator, Negative Testing), and Live (API Calls). The main content area displays the 'My App' details, including the app display name and a 'Generate New Secret' button. Below this, the 'SANDBOX API CREDENTIALS' section shows the 'Sandbox account' (sb-pclks1988345@business.example.com), the 'Client ID' (AYwsmkUm_pRHTqU5GaDFdf-0_NAQwh6aSI9OFzueq9SKPO4whXbtY_ezUvhvNpYjp-F5vEuGFgonClyZ), and the 'Secret' (EJg36tDfAjdiZu-vZkWJXdCS5OzyEiXKvp9f_PdGFzUBhL8WpfsqemhiVU9XPTx mp88QTQsILTk8ImhA). A note states: 'When you generate a new secret, you still maintain the original secret. The maximum number of client secrets is two. A client secret is either in enabled or disabled state.' A table lists the secrets with columns for Created, Secret, Status, and Action. The first secret is 'Enabled' and has an action menu. A 'Generate New Secret' button is at the bottom.

Created	Secret	Status	Action
Jun 03, 2020	EJg36tDfAjdiZu-vZkWJXdCS5OzyEiXKvp9f_PdGFzUBhL8WpfsqemhiVU9XPTx mp88QTQsILTk8ImhA	Enabled	...

4. Tạo tài khoản Sandbox để test

Sau khi get được thông tin xác thực API (API credentials), ta sẽ tiến hành tạo tài khoản cho người mua (buyer) và người bán (merchant/seller) để giao dịch.

PayPal có hai môi trường phục vụ cho việc test (Sandbox Environment) và production (Live Environment).

Environment	Base URL	Test here
Sandbox	https://api.sandbox.paypal.com	www.sandbox.paypal.com
Live	https://api.paypal.com	www.paypal.com

PayPal Sandbox là môi trường để chúng ta có thể phát triển, test thử các giao dịch, đảm bảo việc tích hợp PayPal chính xác trước khi chạy trên production.

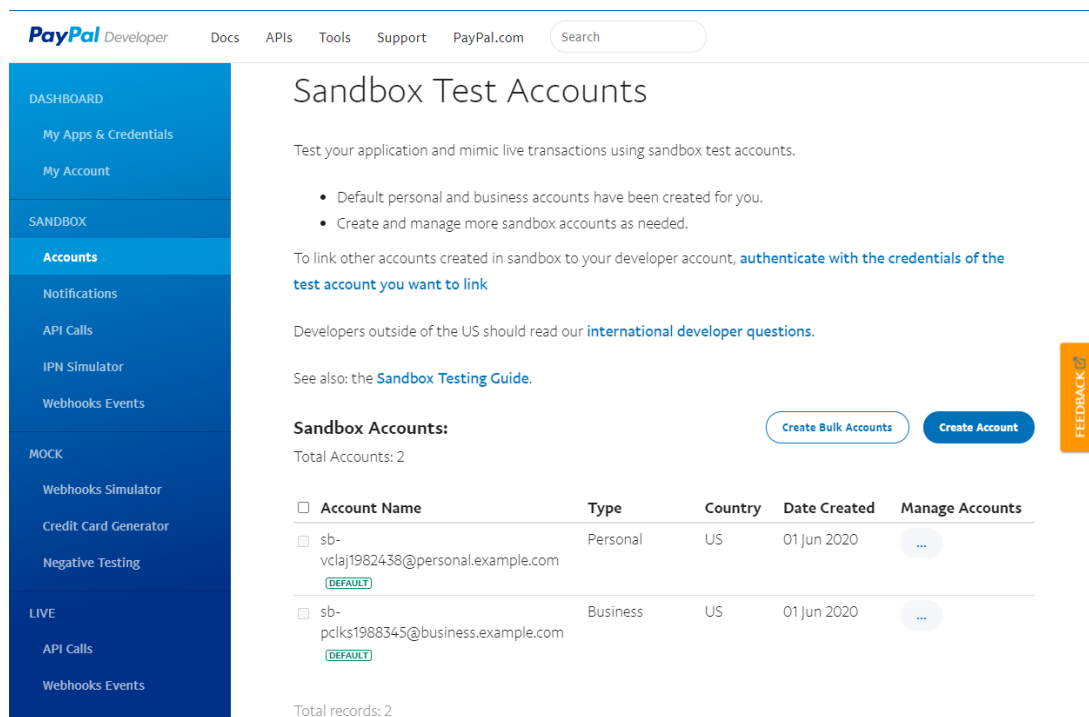
Có hai loại tài khoản:

- Personal: tài khoản cho người mua (bên chuyển tiền)
- Business: tài khoản cho người bán (bên nhận tiền)

Account type	Represents
Personal	The customer in a transaction.
Business	The merchant in a transaction.

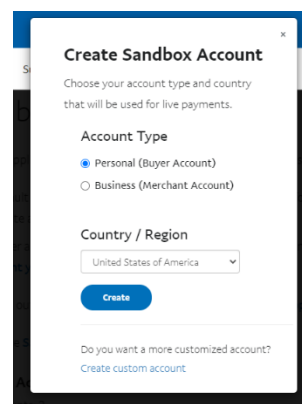
Mặc định thì PayPal đã tạo sẵn 2 tài khoản cho người mua (Personal) và người bán (Business).

Ta có thể sử dụng tài khoản mặc định này để test các giao dịch hoặc có thể tạo thêm.



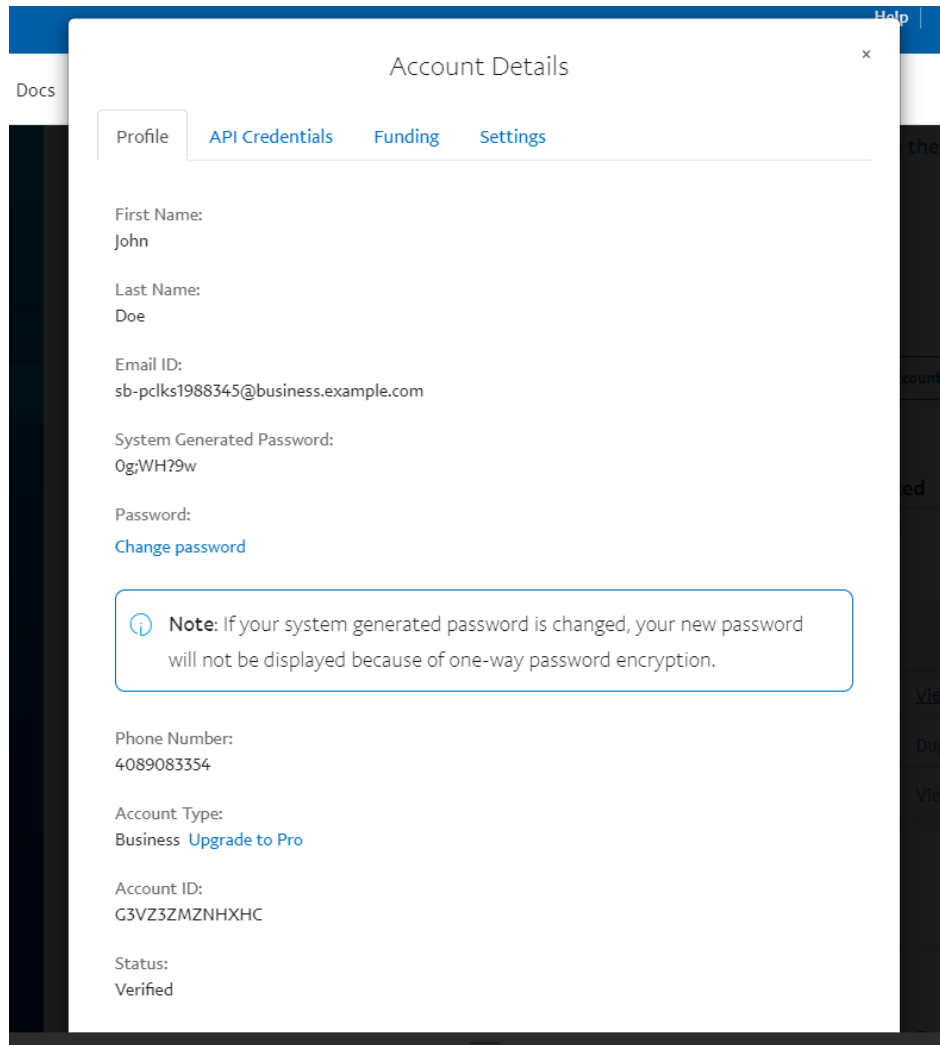
Tạo tài khoản Sanbox:

1. Chọn **Accounts** ở menu Sandbox
2. Chọn **Create Account**
3. Chọn **Account Type** và nhấn **Create**



Đăng nhập vào PayPal Sandbox để xem lịch sử giao dịch, số dư tài khoản:

1. Lựa chọn tài khoản sandbox
2. Chọn Manage Accounts -> View/Edit Account
3. Lấy thông tin đăng nhập bao gồm email và password (system generated password)

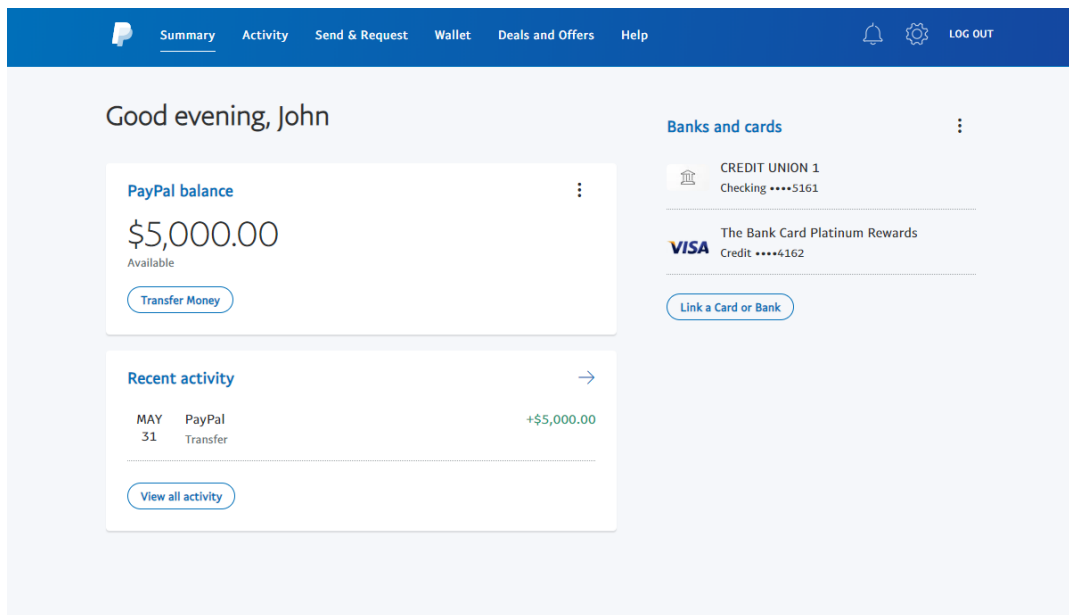


The screenshot shows a modal window titled "Account Details" with a close button (x) in the top right corner. The modal has four tabs: "Profile" (selected), "API Credentials", "Funding", and "Settings". The "Profile" tab displays the following information:

- First Name: John
- Last Name: Doe
- Email ID: sb-pclks1988345@business.example.com
- System Generated Password: 0g:WH79w
- Password: [Change password](#)
- Phone Number: 4089083354
- Account Type: Business [Upgrade to Pro](#)
- Account ID: G3VZ3ZMZNHXHC
- Status: Verified

A note box with an information icon (i) contains the text: "Note: If your system generated password is changed, your new password will not be displayed because of one-way password encryption."

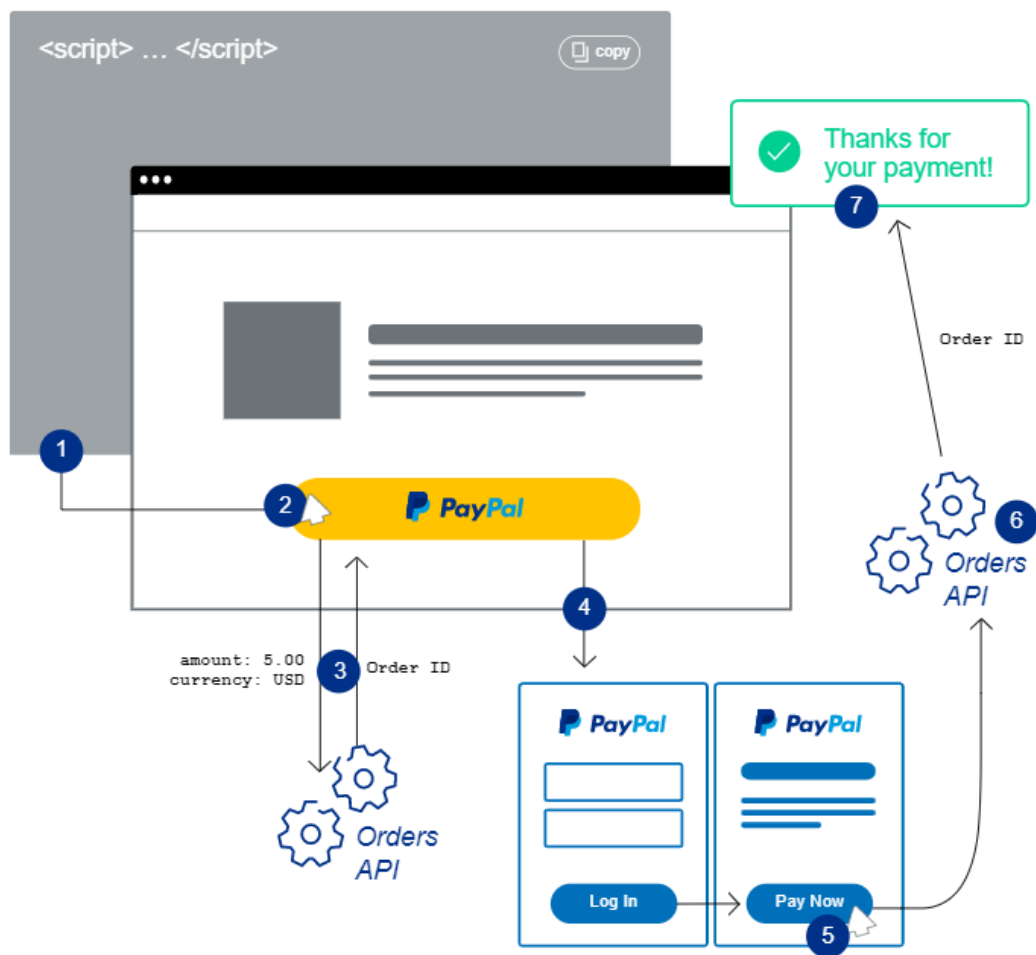
4. Truy cập <https://www.sandbox.paypal.com/>
5. Đăng nhập với email và password ở trên



Mặc định mỗi tài khoản Sandbox sẽ có sẵn 5000\$ để chúng ta có thể giao dịch.

Muốn biết giao dịch thành công hay thất bại (đã chuyển được tiền hay chưa) thì chỉ cần đăng nhập vào tài khoản của người bán và người mua để xem số dư (PayPal balance) và lịch sử giao dịch (Activity).

PayPal Payment Process



Quy trình tích hợp thanh toán với PayPal

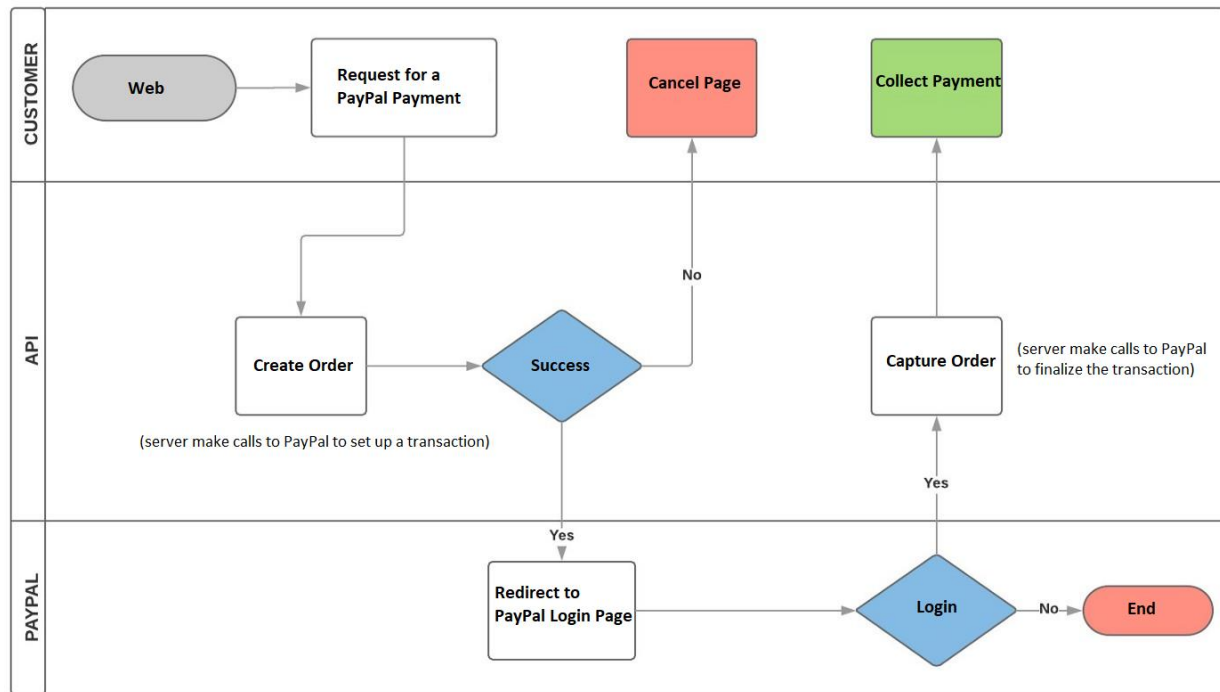
1. Thêm nút thanh toán với PayPal vào web
2. Người mua nhấn vào nút thanh toán với PayPal
3. Gọi PayPal Orders API để tiến hành phiên giao dịch (set up a transaction) (gửi số tiền, đơn vị tiền + các thông tin khác)
4. Hiển thị trang đăng nhập PayPal, người mua đăng nhập vào PayPal và nhấn nút Pay Now
5. Gọi PayPal Orders API để hoàn thành phiên giao dịch (finalize the transaction)
6. Hiển thị trang xác nhận thanh toán thành công đến người mua

Việc call Orders API (hiểu đơn giản là API do PayPal cung cấp để tiến hành giao dịch) có thể thực hiện ở Client-Side hoặc Server-Side.

Call Orders API ở Client-Side giúp việc tích hợp thanh toán với PayPal dễ dàng. Tuy nhiên chúng ta sẽ call Orders API ở phía Server-Side để dễ dàng thao tác với hệ thống, services, database và bảo mật tốt hơn (Client ID và Secret sẽ lưu ở phía backend).

“PayPal provides client-side shortcuts to help you complete a transaction with minimal server integration. You can also make calls to PayPal APIs directly from your server to integrate PayPal more closely with your backend systems, services, and database.”

Integration



Set Up PayPal JavaScript SDK Script

Thêm PayPal JavaScript SDK vào web

```
<!DOCTYPE html>

<head>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
</head>

<body>
  <script src="https://www.paypal.com/sdk/js?client-id=CLIENT_ID">
  </script>
</body>
```

Thay thế `CLIENT_ID` với Sandbox Client Id đã tạo

Xem thêm cách customize PayPal Javascript SDK tại đây:

<https://developer.paypal.com/docs/checkout/reference/customize-sdk/>

Render button PayPal

```
<body>
  <script src="https://www.paypal.com/sdk/js?client-id=CLIENT_ID">
  </script>

  <div id="paypal-button-container"></div>

  <script>
    paypal.Buttons().render('#paypal-button-container');
  </script>
</body>
```

Xem thêm cách customize button tại đây:

<https://developer.paypal.com/docs/checkout/integration-features/customize-button/>

Set Up Server-Side SDK

Ta sẽ không gọi trực tiếp đến API của PayPal mà sẽ dùng thư viện hỗ trợ (hay còn gọi là SDK).

PayPal REST SDKs cung cấp cho chúng ta các model, method cần thiết để có thể sử dụng để gọi đến PayPal API (nếu không sử dụng SDK thì ta sẽ phải tự tạo model, call http, xử lý authentication...)

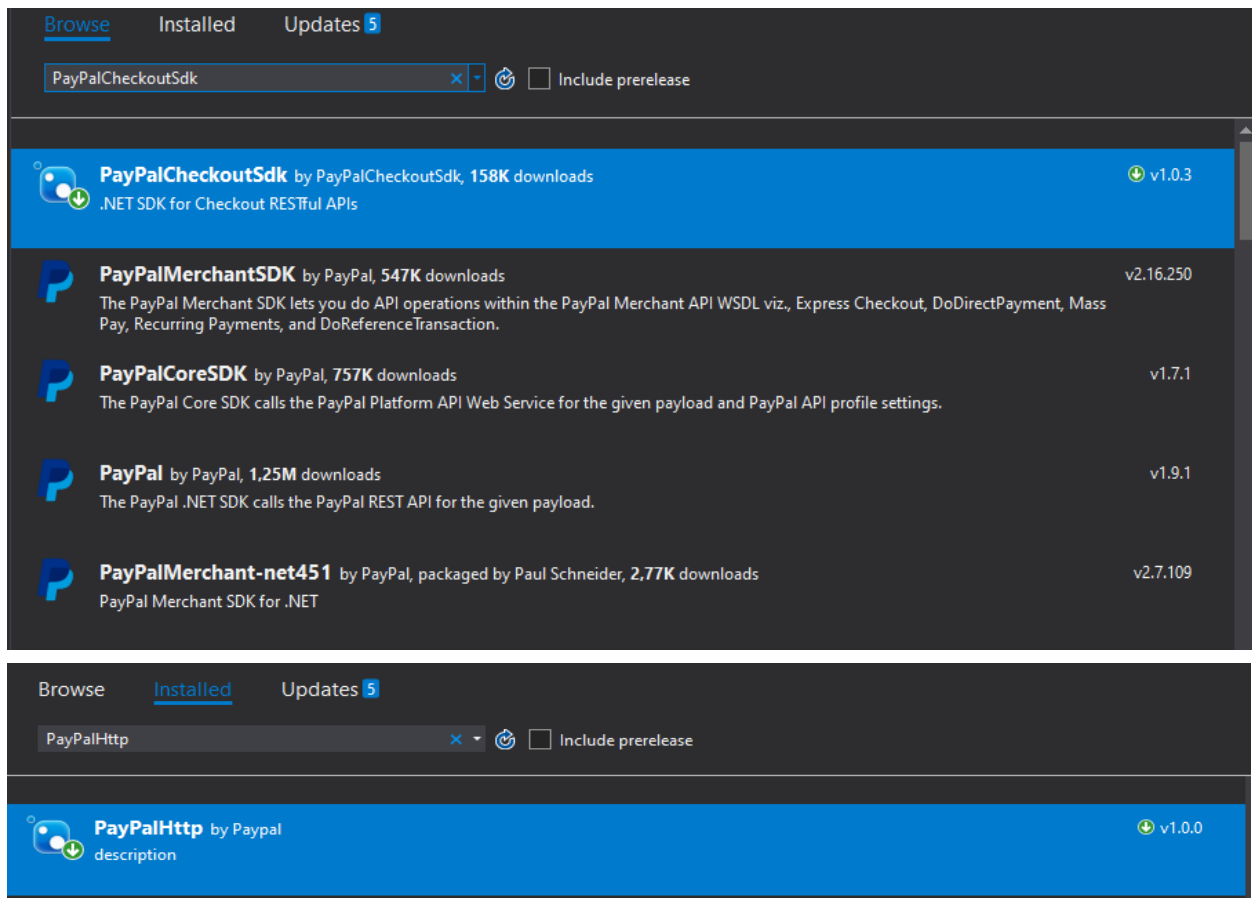
PayPal REST SDKs có sẵn cho: `Java`, `PHP`, `Node`, `Python`, `Ruby`, `.NET`.

Cài đặt SDK cho .NET Core

- Cài đặt qua .NET CLI

```
dotnet add package PayPalCheckoutSdk
dotnet add package PayPalHttp
```

- Cài đặt qua Nuget packages



Sau khi cài đặt SDK thành công, tại project đã cài đặt SDK, tạo file **PayPalClient.cs** để chứa những config bao gồm: config PayPal environment (development, testing dùng Sandbox environment, production dùng Live Enviroment), setup PayPal Credentials bao gồm Client ID, Client Secret.

```
using PayPalCheckoutSdk.Core;
using PayPalHttp;

public class PayPalClient
{
    /**
     * Set up PayPal environment with sandbox credentials.
     * In production, use LiveEnvironment.
     */
    public static PayPalEnvironment Environment()
    {
        return new SandboxEnvironment("CLIENT_ID", "CLIENT_SECRET");
    }

    /**
     * Returns PayPalHttpClient instance to invoke PayPal APIs.
     */
    public static HttpClient Client()
    {
        return new PayPalHttpClient(Environment());
    }
}
```

Thay thế CLIENT_ID và CLIENT_SECRET với Sandbox Client Id và Sandbox Client Secret đã tạo

Set Up Transaction

Ở phía server:

CheckoutController.cs

```
[HttpPost("api/checkout/orders ")]
public async Task<IActionResult> Create()
{
    var order = await checkoutService.CreateOrder();

    return Ok(order);
}
```

CheckoutService.cs

```
public async Task<Order> CreateOrder()
{
    var request = new OrdersCreateRequest();
    request.Prefer("return=representation");
    request.RequestBody(GetOrderRequest());

    // Call PayPal to set up a transaction
    var response = await PayPalClient.Client().Execute(request);

    return response.Result<Order>();
}
```


```

private OrderRequest GetOrderRequest() {
    return new OrderRequest {
        CheckoutPaymentIntent = "CAPTURE",
        ApplicationContext = new ApplicationContext {
            BrandName = "<-Your Brand Name->",
            LandingPage = "NO_PREFERENCE",
            UserAction = "CONTINUE",
            ShippingPreference = "NO_SHIPPING"
        },
        PurchaseUnits = new List<PurchaseUnitRequest> {
            new PurchaseUnitRequest {
                AmountWithBreakdown = new AmountWithBreakdown {
                    CurrencyCode = "USD",
                    Value = "100",
                    AmountBreakdown = new AmountBreakdown {
                        ItemTotal = new Money {
                            CurrencyCode = "USD",
                            Value = "100"
                        }
                    }
                },
                Items = new List<Item> {
                    new Item {
                        Name = "<-Item name->",
                        Description = "<-Item description->",
                        UnitAmount = new Money {
                            CurrencyCode = "USD",
                            Value = "100"
                        },
                        Quantity = "1"
                    }
                }
            }
        }
    };
}

```

Test Checkout API với Postman

POST <https://localhost:44314/api/checkout/orders>

Pretty Raw Preview Visualize JSON 

```
1  {
2    "intent": "CAPTURE",
3    "create_time": "2020-06-08T01:38:12Z",
4    "id": "9JX46216UM3799726",
5    "links": [
6      {
7        "href": "https://api.sandbox.paypal.com/v2/checkout/orders/9JX46216UM3799726",
8        "method": "GET",
9        "rel": "self"
10     },
11     {
12       "href": "https://www.sandbox.paypal.com/checkoutnow?token=9JX46216UM3799726",
13       "method": "GET",
14       "rel": "approve"
15     },
16     {
17       "href": "https://api.sandbox.paypal.com/v2/checkout/orders/9JX46216UM3799726",
18       "method": "PATCH",
19       "rel": "update"
20     },
21     {
22       "href": "https://api.sandbox.paypal.com/v2/checkout/orders/9JX46216UM3799726/capture",
23       "method": "POST",
24       "rel": "capture"
25     }
26   ],
27   "purchase_units": [
28     {
29       "amount": {
30         "breakdown": {
31           "item_total": {
32             "currency_code": "USD",
33             "value": "1000000.00"
34           }
35         },
36         "currency_code": "USD",
37         "value": "1000000.00"
38       },
39       "items": [
40         {
41           "description": "<-Item description->",
42           "name": "<-Item name->",
43           "quantity": "1",
44           "unit_amount": {
45             "currency_code": "USD",
46             "value": "1000000.00"
47           }
48         }
49       ],
50       "payee": {
51         "email": "test@paypal.com",
52         "first_name": "John",
53         "last_name": "Doe",
54         "phone": "1234567890",
55         "country_code": "US",
56         "postal_code": "90210",
57         "city": "Beverly Hills",
58         "state": "CA",
59         "zip": "90210"
60       }
61     }
62   ]
63 }
```

Ở phía client:

```
<script>
  paypal.Buttons({
    createOrder: function () {
      return fetch('/my-server/api/checkout/orders', {
        method: 'post',
        headers: {
          'content-type': 'application/json'
        }
      }).then(function (res) {
        return res.json();
      }).then(function (data) {
        return data.id;
      });
    }

  }).render('#paypal-button-container');
</script>
```


Capture Transaction

Ở phía server:

CheckoutController.cs

```
[HttpPost("api/checkout/orders/{id}/capture")]
public async Task<IActionResult> Capture([FromRoute] string id)
{
    var order = await checkoutService.CaptureOrder(id);

    return Ok(order);
}
```

CheckoutService.cs

```
public async Task<Order> CaptureOrder(string orderId)
{
    var request = new OrdersCaptureRequest(orderId);
    request.Prefer("return=representation");
    request.RequestBody(new OrderActionRequest());

    // Call PayPal to capture an order
    var response = await PayPalClient.Client().Execute(request);

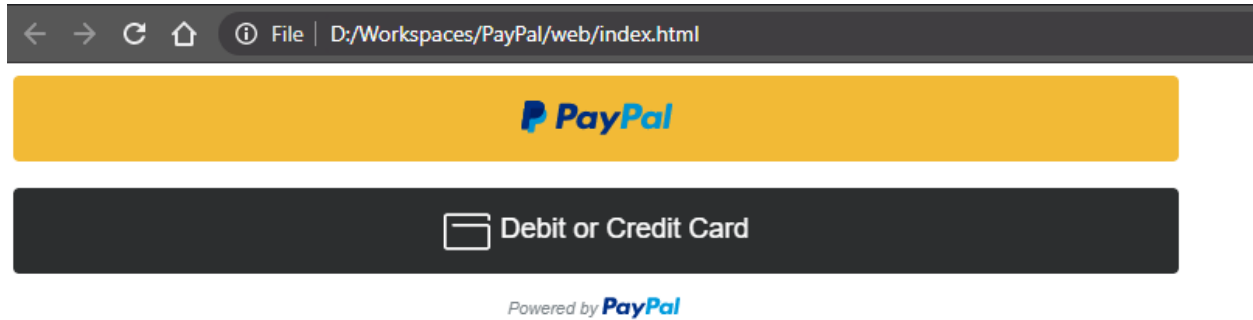
    return response.Result<Order>();
}
```

Ở phía client:

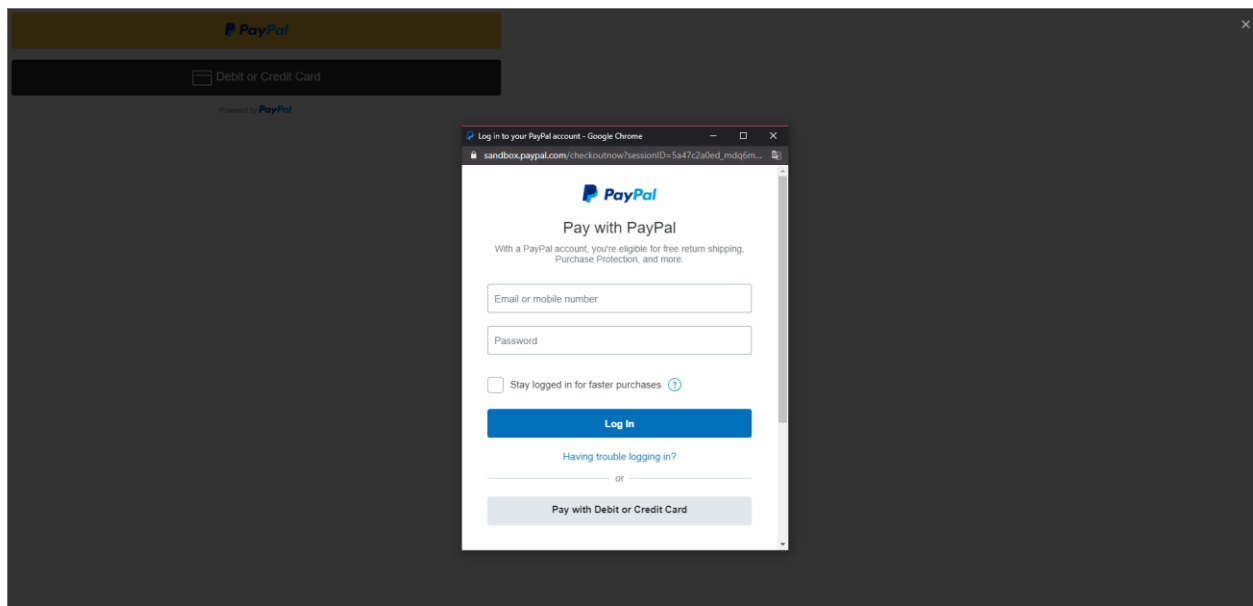
```
<script>
  paypal.Buttons({
    createOrder: function () {
      return fetch('/my-server/api/checkout/orders', {
        method: 'post',
        headers: {
          'content-type': 'application/json'
        }
      }).then(function (res) {
        return res.json();
      }).then(function (data) {
        return data.id;
      });
    },
    onApprove: function (data) {
      return fetch(`/my-server/api/checkout/orders/${data.orderID}/capture`, {
        method: 'post',
        headers: {
          'content-type': 'application/json'
        }
      }).then(function (res) {
        return res.json();
      }).then(function (details) {
        alert('Transaction funds captured');
      })
    }
  }).render('#paypal-button-container');
</script>
```

Testing

PayPal button



Click PayPal -> Đăng nhập với Personal Account




Sau khi đăng nhập thành công PayPal -> Click Pay Now

PayPal Checkout - Google Chrome

sandbox.paypal.com/webapps/hermes?flow=1-P&ulReturn=true&sessionID=5a47c2a0ed_mdq6mdy6mtk&buttonSessionID=a869a4df34_mdq6mdy6mtk&...


<-Your Brand Name->

\$100.00 USD ^


1 <-Item name->	\$100.00 USD
<-Item description->	
Subtotal	\$100.00 USD
Total	\$100.00 USD


Hi, John!

Pay with

☒  Balance \$100.00 USD

☐ Make this my preferred way to pay

☐  The Bank Card Platinum Rewards
Credit ****4162

☐  CREDIT UNION 1
Checking ****5161

[+ Add debit or credit card](#)

[+ !\[\]\(68b7531d6e371d09b205fb8e156d145c_img.jpg\) Apply for PayPal Credit](#)

No Interest if paid in full in 6 months for your purchase of \$100.00.
Subject to credit approval. [See terms](#)

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

Pay Now

[Cancel and return to <-Your Brand Name->](#)

© 1999 - 2020 [Legal](#) [Privacy](#) [Feedback](#)

Đăng nhập Sandbox với Personal Account

Good evening, John

PayPal balance

\$4,900.00

Available

Transfer Money

Recent activity

JUN 04	John Doe's Test Store Payment	-\$100.00
MAY 31	PayPal Transfer	+\$5,000.00

View all activity


Đăng nhập Sandbox với Business Account

PayPal Balance

\$5,096.80 USD

Available


Transfer Money

 **Recent activity**
PayPal Activity

[Payments received](#) [Payments sent](#) [Activity \(including balance & fees\)](#)

9:28 PM	Payment from John Doe Completed	\$100.00 USD
Apr 3, 2020	Transfer from PayPal Completed	\$5,000.00 USD

[View all](#)

Transaction details  **Print**

Payment received from John Doe
June 4, 2020 at 9:28:04 PM PDT
Payment Status: **COMPLETED**
Payment Type: Checkout

Gross amount
\$100.00 USD

Transaction ID: 82966277114094313

[Print packing slip](#) | [Add tracking info](#) | [Mark as shipped](#)

We have no postal address on file

Order details	Quantity	Price	Subtotal
	1	\$100.00 USD	\$100.00 USD
		Purchase Total	\$100.00 USD

Payment details

Purchase Total	\$100.00 USD
Sales Tax	\$0.00 USD
Shipping Amount	\$0.00 USD
Handling Amount	\$0.00 USD
Insurance Amount	\$0.00 USD
Gross Amount	\$100.00 USD
PayPal Fee	-\$3.20 USD
Net Amount	\$96.80 USD

[Issue a refund](#)

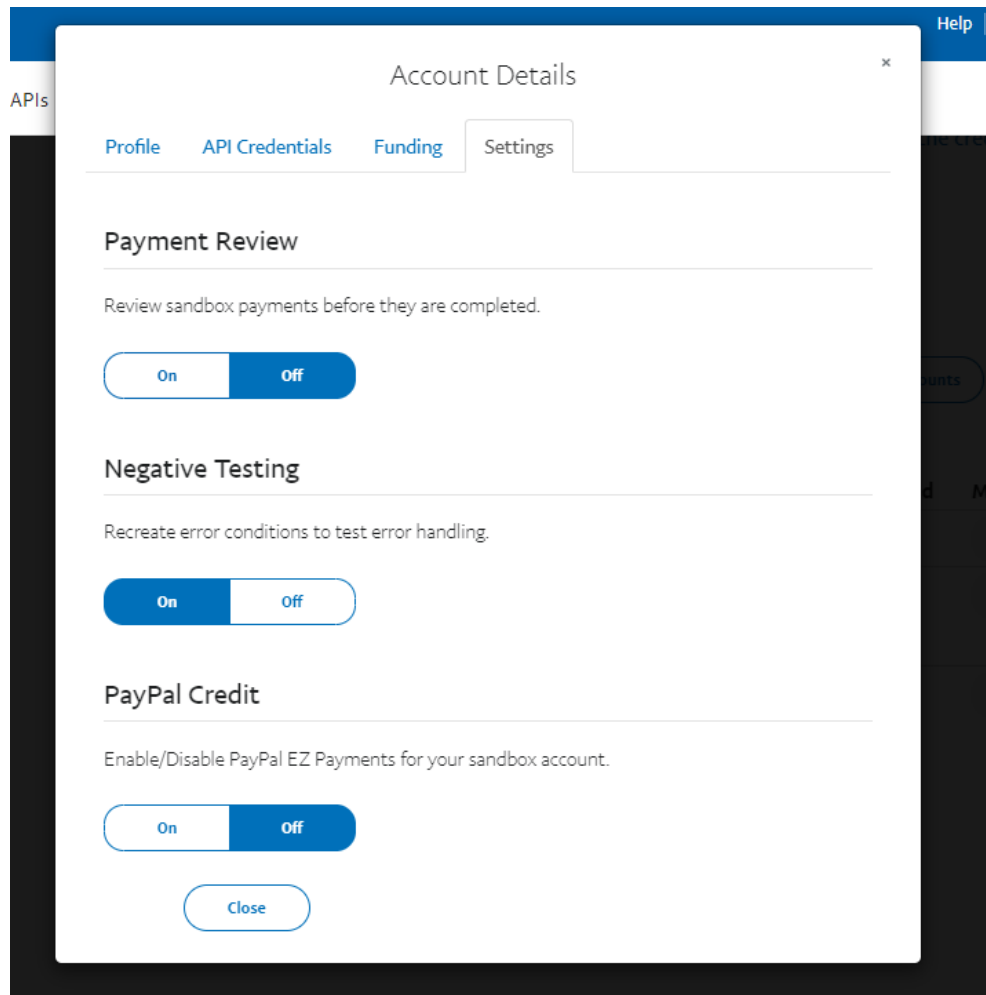
Reversals – Please be aware that this payment can still be reversed (e.g. if it is subject to a chargeback), even after you have sent the item(s) to your buyer. Complying with PayPal's Seller Protection and following the trading guidelines on our Security page helps to protect you from things like chargebacks.

Refund Information – Please accept or refuse this payment. If you accept this payment now and need to refund it later, the Issue a Refund link will be available on the transaction details page for 180 days after the payment was originally sent.

Test các trường hợp hợp lỗi với negative testing

Bật negative testing

1. Truy cập <https://developer.paypal.com> và đăng nhập vào Dashboard với tài khoản PayPal Developer
2. Chọn **Accounts** ở menu **Sandbox**
3. Chọn tài khoản **Business Sandbox** đã tạo
4. Tại cột **Manage Account** -> **View/Edit Account**
5. Chọn **Settings** tab
6. Tại option **Negative Testing** -> click **ON**



Test với negative testing

REST API apps dùng request header để thực thi negative testing ở sandbox.

"REST API apps use a request header to invoke negative testing in the sandbox. This header configures the sandbox into a negative testing state for transactions that include the merchant."

Thêm key `PayPal-Mock-Response` vào request header khi gọi PayPal API.

Ở Server:

CheckoutService.cs

```
public async Task<Order> CaptureOrder(string orderId)
{
    var request = new OrdersCaptureRequest(orderId);
    request.Prefer("return=representation");
    request.Headers.Add("PayPal-Mock-Response", "{\n\"mock_application_codes\": \"DUPLICATE_INVOICE_ID\"}");
    request.RequestBody(new OrderActionRequest());

    // Call PayPal to capture an order
    var response = await PayPalClient.Client().Execute(request);

    return response.Result<Order>();
}
```


CheckoutController.cs

```
[HttpPost("api/checkout/orders/{id}/capture")]
public async Task<IActionResult> Capture([FromRoute] string id)
{
    try
    {
        var order = await checkoutService.CaptureOrder(id);

        return Ok(new
        {
            Success = true,
            Result = order
        });
    }
    catch (Exception exception)
    {
        return BadRequest(new
        {
            Success = false,
            Error = JsonConvert.DeserializeObject(exception.Message)
        });
    }
}
```

Ö Client:

```
<script>
  paypal.Buttons({
    onApprove: function (data, actions) {
      return fetch(`/my-server/api/checkout/orders/${data.orderID}/capture`, {
        method: 'post',
        headers: {
          'content-type': 'application/json'
        }
      }).then(function (res) {
        return res.json();
      }).then(function (details) {
        console.log(details);
/* If your buyer's funding source fails,
 * the Orders API returns an INSTRUMENT_DECLINED error.
 * A funding source might fail for these reasons:
 * - The billing address associated with the financial instrument could not be
 * confirmed.
 * - The transaction exceeds the card limit.
 * - The card issuer denied the transaction.
 * */
        if (details.error && details.error.name === 'INSTRUMENT_DECLINED') {
          return actions.restart();
        }
      })
    },
/* When a buyer cancels a payment, they typically return to the parent page.
 * You can instead use the onCancel function to show a cancellation page
 * or return to the shopping cart
 * */
    onCancel: function (data) {
      // Show a cancel page, or return to cart
    },
/* If an error prevents buyer checkout,
 * show an error page using the onError callback
 * */
    onError: function (err) {
      // Show an error page here, when an error occurs
    }
  }).render('#paypal-button-container');
</script>
```

```
✖ ▶ POST https://localhost:44314/api/checkout/orders/0TY7714208043394P/capture 400
▼ {success: false, error: {…}} ⓘ
  error:
    debug_id: "70c28ae654da"
    details: Array(1)
      ▼ 0:
        description: "Duplicate Invoice ID detected. To avoid a potential duplicate transaction your account setting requires that Invoice Id be unique for each transaction."
        issue: "DUPLICATE_INVOICE_ID"
        __proto__: Object
        length: 1
        __proto__: Array(0)
    links: Array(1)
      ▼ 0: {href: "https://developer.paypal.com/docs/api/orders/v2/#error-DUPLICATE_INVOICE_ID", rel: "information_link", method: "GET"}
        length: 1
        __proto__: Array(0)
    message: "The requested action could not be completed, was semantically incorrect, or failed business validation."
    name: "UNPROCESSABLE_ENTITY"
    __proto__: Object
    success: false
    __proto__: Object
```

Custom credit and debit card payments

Subscriptions

With PayPal Subscriptions, you can bill customers for physical and digital goods or services at regular intervals.

Integration features

Use the Subscriptions API or Subscriptions Dashboard to:

- Create plans that charge users a fixed amount at regular intervals, or based on the quantity to which they subscribe.
- Offer your subscribers free or discounted trials to get more subscription signups.
- Easily modify the price of your plans.
- Accept currencies for over 250 countries.
- Enable subscribers to upgrade and downgrade their plans or change the quantity to which they subscribe.
- Automate payment recovery for failed payments.

<https://developer.paypal.com/docs/subscriptions>