

## Thanh toán tích hợp Paypal

Tham khảo bài viết gốc trên code project :

<https://www.codeproject.com/Articles/870870/Using-Paypal-Rest-API-with-ASP-NET-MVC>

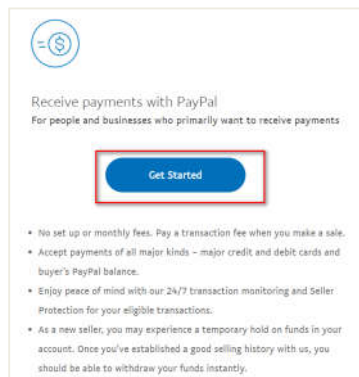
Bài viết có tham khảo và điều chỉnh bỏ bớt một số nội dung không cần thiết, cũng như hướng dẫn và giải thích chi tiết việc tạo tài khoản paypal và môi trường tích hợp để kiểm thử sandbox

### **Nội dung bổ sung thêm:**

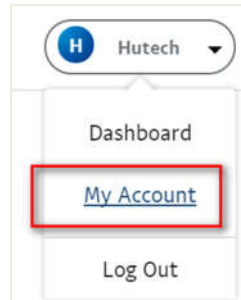
- Hướng dẫn đăng ký tài khoản tích hợp thử trên môi trường sandbox
- Hướng dẫn chuyển từ môi trường test sandbox sang môi trường thanh toán thật trên paypal
- Hướng dẫn cách lấy ClientId và ClientSecret key

#### 1.1 Đăng ký tài khoản tích hợp cho nhà phát triển

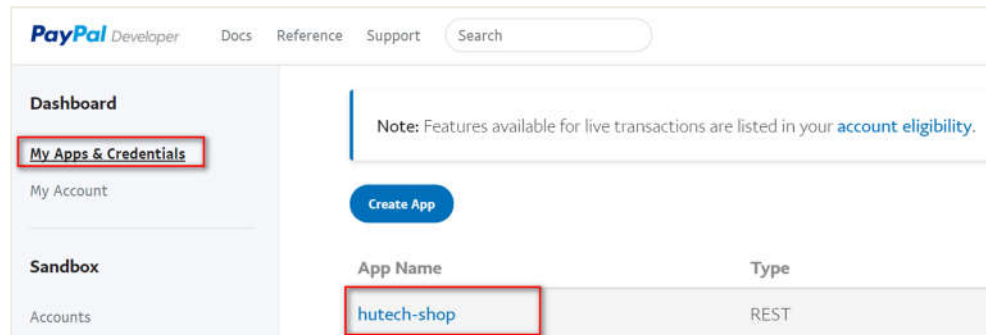
- <https://developer.paypal.com>



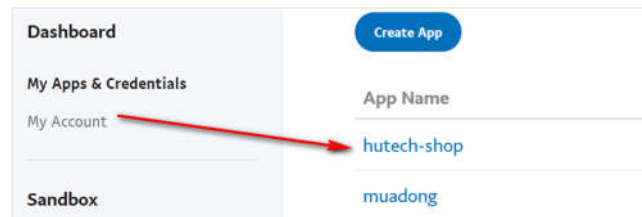
- Xác nhận email
- Đăng ký tích hợp ứng dụng vào website



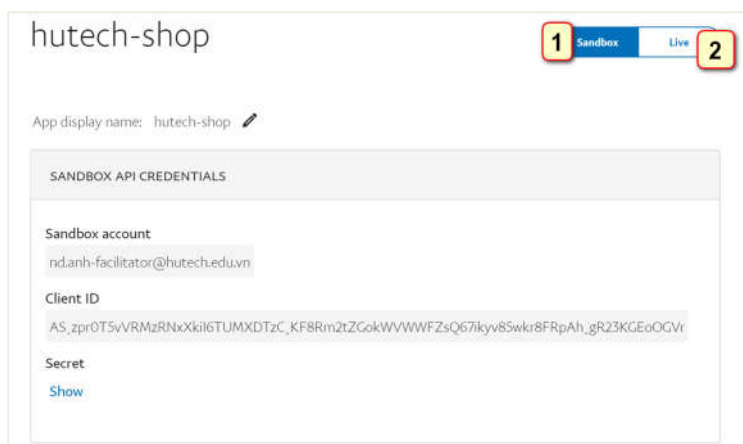
- Lấy thông tin Client ID và Client Secret key



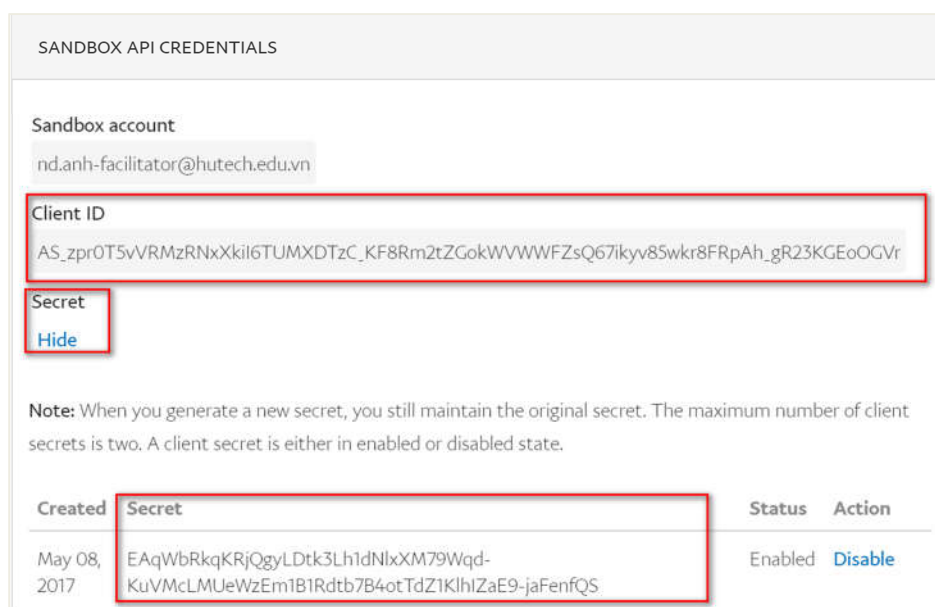
- **Hiểu về tài khoản sandbox:** paypal tạo môi trường thanh toán ảo tại **sandbox.paypal.com** cho nhà phát triển gần giống với môi trường thanh toán và quản lý tiền thật **paypal.com**, nhà phát triển cần phải kiểm tra kỹ lưỡng trước khi đưa ra môi trường triển khai thật.



- Vào App



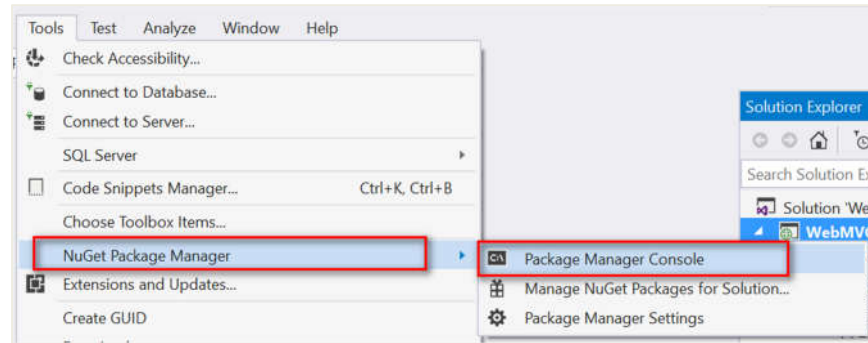
- [1] Môi trường kiểm thử tích hợp ảo cho nhà phát triển
- [2] Môi trường tích hợp thật



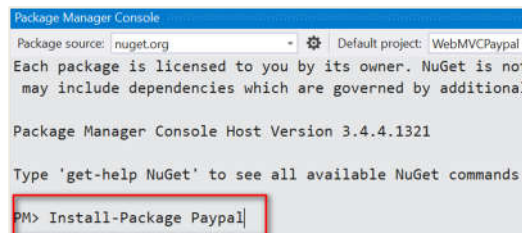
- Lưu **Client ID** và **Client Secret key** vào file text document (txt) để làm ở bước sau

## 1.2 Tích hợp vào ứng dụng website (lưu ý: bước này đã có sẵn website bán hàng cần tích hợp thanh toán paypal hoặc tự tạo ứng dụng website asp net mvc để thực hiện)

- Tạo mới project ASP NET MVC Web Application → đặt tên ứng dụng.
- Thêm thư viện **Paypal SDK for .NET** thông qua **NuGet Package Manager** hoặc thêm từ viện PayPal.dll, Newtonsoft.Json.dll từ thư mục <<thư-vien>> kèm theo bài hướng dẫn
- Chọn Tools -> Library Package Manager -> Package Manager Console



- Install-Package Paypal



### 1.3 Cấu hình file web.config bao gồm các thông tin thanh toán từ paypal

- **Client ID và Client Secret key** lấy ở bước trên (môi trường sandbox)

```
<configSections>
  <!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/?LinkID=237468 -->
  <section name="entityFramework"
    type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework,
    Version=6.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" requirePermission="false" />
  <section name="paypal" type="PayPal.SDKConfigHandler, PayPal" />
</configSections>
```

- Code

```
<section name="paypal" type="PayPal.SDKConfigHandler, PayPal" />
```

- Tiếp theo

```

<configuration>
  <configSections>...</configSections>
  <connectionStrings>...</connectionStrings>
  <appSettings>...</appSettings>
  <system.web>...</system.web>
  <system.webServer>...</system.webServer>
  <runtime>...</runtime>
  <entityFramework>...</entityFramework>
  <system.codedom>...</system.codedom>
  <!-- PayPal SDK settings -->
  <paypal>
    <settings>
      <add name="mode" value="sandbox"/>
      <add name="connectionTimeout" value="360000"/>
      <add name="requestRetries" value="1"/>
      <add name="clientId"
        value="AS_zpr0T5vVRMzRNxXkiI6TUMXDTzC_KF8Rm2tZGokWVWWFZsQ67ikyv85wkr8FRp
        r"/>
      <add name="clientSecret" value="EAqWbRkqKRjQgyLDtk3Lh1dNlxXM79Wqd-
        KuVMcLMUewzEm1B1Rdtb7B4otTdZ1K1hIZaE9-jaFenfQS"/>
    </settings>
  </paypal>
</configuration>

```

Sandbox chỉ là môi trường kiểm thử, trước khi triển khai ứng dụng thanh toán trên paypal cần đổi mode "live" và thay đổi clientId và clientSecret key

- Code

```

<paypal>
  <settings>
    <add name="mode" value="sandbox" />
    <add name="connectionTimeout" value="360000" />
    <add name="requestRetries" value="1" />
    <add name="clientId" value="Giá trị clientId" />
    <add name="clientSecret" value="Giá trị Secret" />
  </settings>
</paypal>

```

#### 1.4 Tạo controller thực hiện thanh toán qua **Paypal** đặt tên **PaypalController.cs**

```
namespace WebMVCPaypal.Controllers
{
    public class PaypalController : Controller
    {
        private PayPal.Api.Payment payment;
        // GET: Paypal
        public ActionResult Index()
        {
            return View();
        }

        public ActionResult PaymentWithPaypal()...

        private Payment CreatePayment(APIContext apiContext, string returnUrl)...

        private Payment ExecutePayment(APIContext apiContext, string payerId, string
            paymentId)...
    }
}
```

```
public ActionResult PaymentWithPaypal()
{
    //getting the apiContext as earlier
    APIContext apiContext = Configuration.GetAPIContext();

    try
    {
        string payerId = Request.Params["PayerID"];

        if (string.IsNullOrEmpty(payerId))
        {
            //this section will be executed first because PayerID doesn't exist
            //it is returned by the create function call of the payment class

            // Creating a payment
            // baseUrl is the url on which paypal sendsback the data.
            // So we have provided URL of this controller only
            string baseURI = Request.Url.Scheme + "://" + Request.Url.Authority
+
                "/Paypal/PaymentWithPayPal?";

            //guid we are generating for storing the paymentID received in
session
            //after calling the create function and it is used in the payment
execution

            var guid = Convert.ToString((new Random()).Next(100000));

            //CreatePayment function gives us the payment approval url
            //on which payer is redirected for paypal account payment

            var createdPayment = this.CreatePayment(apiContext, baseURI +
```

```

"guid=" + guid);

//get links returned from paypal in response to Create function
call
    var links = createdPayment.links.GetEnumerator();

    string paypalRedirectUrl = null;

    while (links.MoveNext())
    {
        Links lnk = links.Current;

        if (lnk.rel.ToLower().Trim().Equals("approval_url"))
        {
            //saving the paypalredirect URL to which user will be
            redirected for payment
            paypalRedirectUrl = lnk.href;
        }
    }

    // saving the paymentID in the key guid
    Session.Add(guid, createdPayment.id);

    return Redirect(paypalRedirectUrl);
}
else
{
    // This section is executed when we have received all the payments
    parameters

    // from the previous call to the function Create

    // Executing a payment

    var guid = Request.Params["guid"];

    var executedPayment = ExecutePayment(apiContext, payerId,
    Session[guid] as string);

    if (executedPayment.state.ToLower() != "approved")
    {
        return View("FailureView");
    }
}
}
catch (Exception ex)
{
    Logger.Log("Error" + ex.Message);
    return View("FailureView");
}

return View("SuccessView");
}

```

```
private Payment CreatePayment(APIContext apiContext, string redirectUrl)
```

```
{  
  
    var itemList = new ItemList() { items = new List<Item>() };  
    //Các giá trị bao gồm danh sách sản phẩm, thông tin đơn hàng  
    //Sẽ được thay đổi bằng hành vi thao tác mua hàng trên website  
  
    itemList.items.Add(new Item()  
    {  
        //Thông tin đơn hàng  
        name = "Item Name",  
        currency = "USD",  
        price = "5",  
        quantity = "1",  
        sku = "sku"  
    });  
    //Hình thức thanh toán qua paypal  
    var payer = new Payer() { payment_method = "paypal" };  
  
    // Configure Redirect Urls here with RedirectUrls object  
    var redirUrls = new RedirectUrls()  
    {  
        cancel_url = redirectUrl,  
        return_url = redirectUrl  
    };  
    //các thông tin trong đơn hàng  
    var details = new Details()  
    {  
        tax = "1",  
        shipping = "1",  
        subtotal = "5"  
    };  
    //Đơn vị tiền tệ và tổng đơn hàng cần thanh toán  
    var amount = new Amount()  
    {  
        currency = "USD",  
        total = "7", // Total must be equal to sum of shipping, tax and  
        subtotal.  
        details = details  
    };  
  
    var transactionList = new List<Transaction>();  
    //Tất cả thông tin thanh toán cần đưa vào transaction  
    transactionList.Add(new Transaction()  
    {  
        description = "Transaction description.",  
        invoice_number = "your invoice number",  
        amount = amount,  
        item_list = itemList  
    });  
  
    this.payment = new Payment()  
    {  
        intent = "sale",  
        payer = payer,  
        transactions = transactionList,  
        redirect_urls = redirUrls  
    };  
};
```



```
        // Create a payment using a APIContext
        return this.payment.Create(apiContext);
    }
```

```
private Payment ExecutePayment(APIContext apiContext, string payerId, string paymentId)
{
    var paymentExecution = new PaymentExecution() { payer_id = payerId };
    this.payment = new Payment() { id = paymentId };
    return this.payment.Execute(apiContext, paymentExecution);
}
```

## 1.5 Tạo thư mục /helper, thêm 2 file [Configuration.cs](#) và [Logger.cs](#)

### - Configuration.cs

```
public static class Configuration
{
    public readonly static string ClientId;
    public readonly static string ClientSecret;

    // Static constructor for setting the readonly static members.
    static Configuration()
    {
        var config = GetConfig();
        ClientId = config["clientId"];
        ClientSecret = config["clientSecret"];
    }

    // Create the configuration map that contains mode and other optional
    // configuration details.
    public static Dictionary<string, string> GetConfig()
    {
        return PayPal.Api.ConfigManager.Instance.GetProperties();
    }

    // Create accessToken
    private static string GetAccessToken()
    {
        // ###AccessToken
        // Retrieve the access token from
        // OAuthTokenCredential by passing in
        // ClientID and ClientSecret
        // It is not mandatory to generate Access Token on a per call basis.
        // Typically the access token can be generated once and
        // reused within the expiry window
        string accessToken = new OAuthTokenCredential(ClientId, ClientSecret,
            GetConfig()).GetAccessToken();
        return accessToken;
    }

    // Returns APIContext object
    public static APIContext GetAPIContext()
    {
        // ### Api Context
    }
```

```
// Pass in a `APIContext` object to authenticate
// the call and to send a unique request id
// (that ensures idempotency). The SDK generates
// a request id if you do not pass one explicitly.
APIContext apiContext = new APIContext(GetAccessToken());
apiContext.Config = GetConfig();

// Use this variant if you want to pass in a request id
// that is meaningful in your application, ideally
// a order id.
// String requestId = Long.toString(System.nanoTime());
// APIContext apiContext = new APIContext(GetAccessToken(), requestId );

return apiContext;
}
}
```

#### - Logger.cs

```
public class Logger
{
    public static string LogDirectoryPath = Environment.CurrentDirectory;


    public static void Log(String lines)
    {
        // Write the string to a file.append mode is enabled so that the log
        // lines get appended to test.txt than wiping content and writing the log

        try
        {
            System.IO.StreamWriter file = new
System.IO.StreamWriter(LogDirectoryPath + "\\Error.log", true);
            file.WriteLine(DateTime.Now.ToString("dd/MM/yyyy HH:mm:ss") + " --> " +
lines);
            file.Close();
        }
        catch
        {
        }
    }
}
```

### 1.6 Chạy thử ứng dụng tích hợp

- [Http:// localhost:60099/paypal/PaymentWithPaypal](http://localhost:60099/paypal/PaymentWithPaypal)

test facilitator's Test Store

🛒 \$7.00 USD ▾


Pay with PayPal

nd.anh-buyer@hutech.edu.vn

.....

☐ Stay logged in for faster checkout [?](#)

Log In



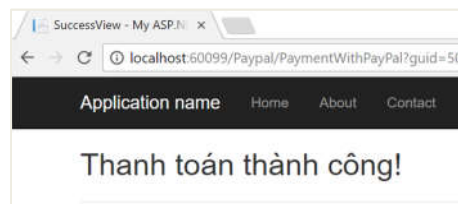
New. Faster. Easier.

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

Tài khoản buyer --> đăng nhập  
<https://developer.paypal.com/> vào  
Sandbox -> Account

- Xử lý

- Kết quả



- Đăng nhập vào <https://www.sandbox.paypal.com/> để kiểm tra đơn hàng

Tài khoản test sandbox --> đăng nhập  
<https://developer.paypal.com> --> Sandbox/ Account



nd.anh-facilitator@hutech.edu.vn

.....

Log In

[Having trouble logging in?](#)

- Xem thống kê đơn hàng

PayPal Summary Money **Activity** Reports Tools ▾ More ▾

Activity Activity (including balance & fees) Charts & Graphs Email Address ▾ Search

Active ▾ All transactions ▾ All currencies ▾ Past 30 days ▾

<input type="checkbox"/>	Date	Type	Name	Payment	Gross	Fee	Net	Balance
<input type="checkbox"/>	5:59 AM	Payment from	test buyer	Completed	\$7.00 USD	-\$0.50	\$6.50	\$13.00 USD
<input type="checkbox"/>	1:04 AM	Payment from	test buyer	Completed	\$7.00 USD	-\$0.50	\$6.50	\$6.50 USD

- Sau khi Kiểm thử thành công trong môi trường Sandbox, chuyển đổi thành môi trường thật để thanh toán paypal (sinh viên tự thực hiện theo mô tả ở phần trên)

## 1.7 Bổ sung: Sinh viên hiện thực các màn hình

- Thiết kế màn hình xem giỏ hàng, người dùng chọn hình thức thanh toán qua paypal

The screenshot shows a shopping cart interface. At the top, there is a link to "Mua thêm sản phẩm khác" and a label "Giỏ hàng của bạn". The main item is an "Apple Macbook Air MGF2ZP/A i5 5250U/8GB/128GB" priced at 23.990.000đ. Below the item name, there are two bullet points: "Tặng Balo thegloiddong (Nếu không có quà sẽ hoàn 100,000đ)" and "Trả góp 0% với HSBC HOẶC Tặng PMH 500,000đ". There is a "Xóa" button and a color selection "Màu: Bạc" with a quantity selector set to 2. The total price is 47.980.000đ, with a note "Sử dụng mã giảm giá". Below this, there are radio buttons for "Anh" (selected) and "Chị". There are input fields for "Tên" (filled with "Nguyễn Đình Ánh") and "Số điện thoại" (filled with "phone"). A red error message "Số điện thoại ít nhất là 10 số" is shown below the phone number field. Below the input fields, there is a prompt "Để được phục vụ nhanh hơn, hãy chọn thêm:" with two radio buttons: "Địa chỉ giao hàng" (selected) and "Nhận tại siêu thị". At the bottom, there are two buttons: "THANH TOÁN KHI NHẬN HÀNG" (with subtext "Xem hàng trước, không mua không sao") and "THANH TOÁN ONLINE" (with subtext "Bảng thẻ ATM, Visa, Master").

- Hệ thống nhận các giá trị của sản phẩm từ CSDL và thông tin khách hàng để thanh toán online qua paypal như hướng dẫn tại các phần trước.