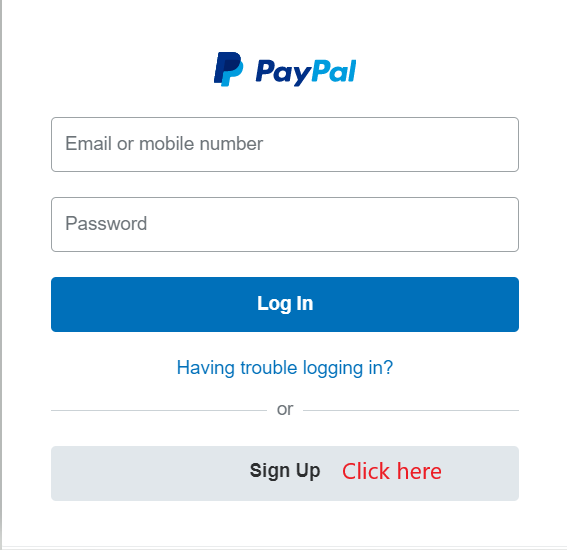
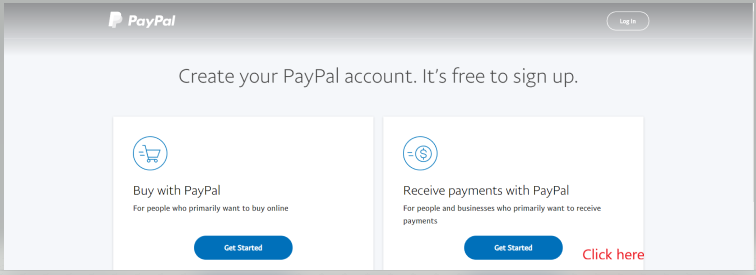
PayPal Gateway Payment

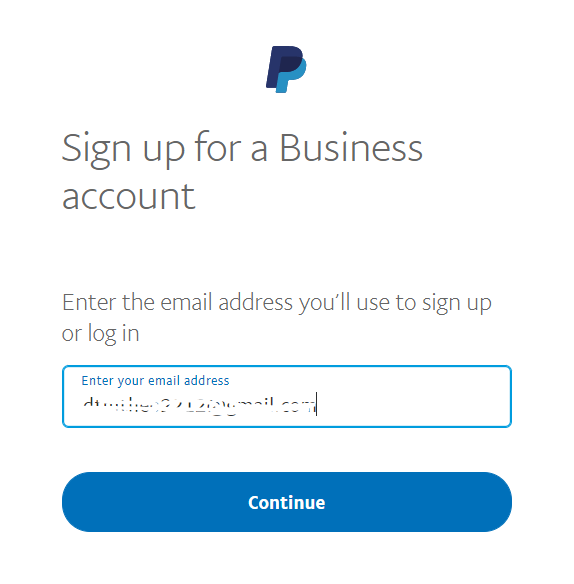
Programming Language: C#

**Step 1:** Create account sandbox developer:

<https://www.paypal.com/signin?returnUri=https%3A%2F%2Fdeveloper.paypal.com%2Fdeveloper%2Fapplications>



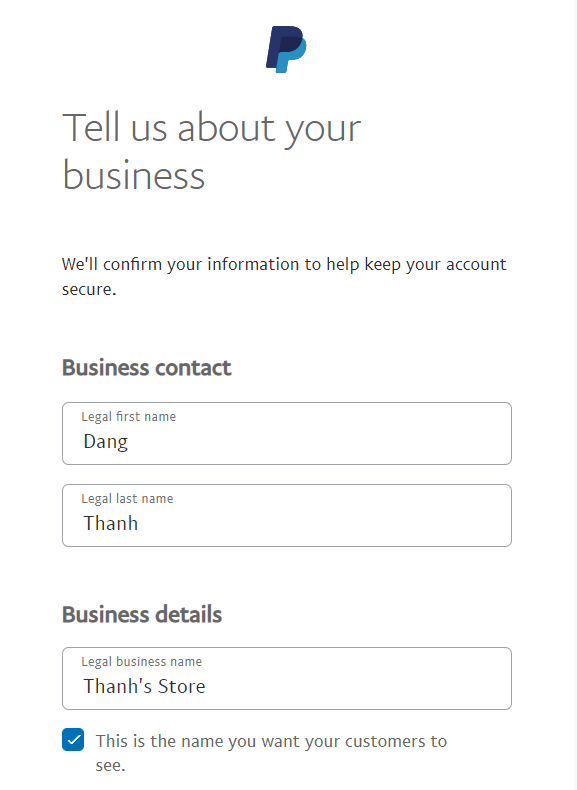
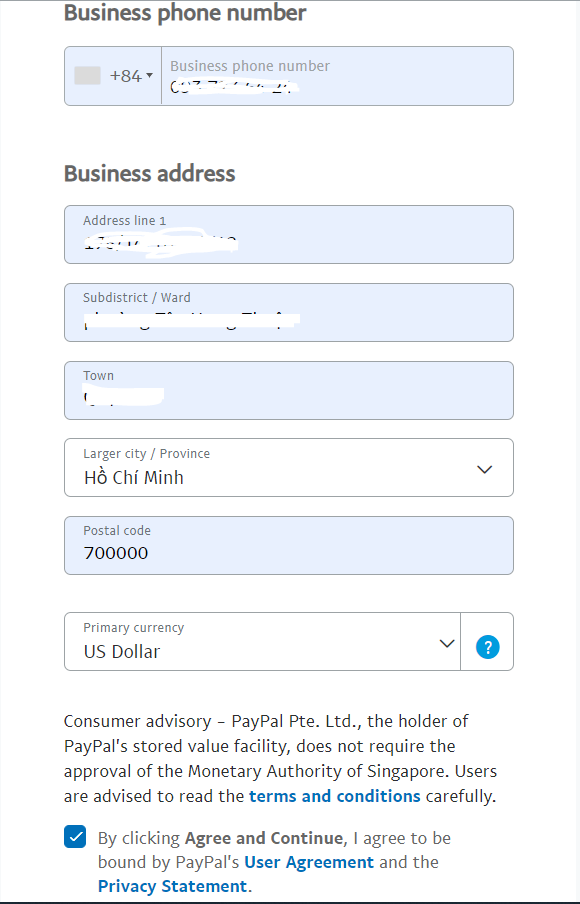


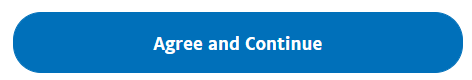


Press the Continue button if you enter your email.

Then enter a password you want and press the Continue button.

Fill in the information that they require.



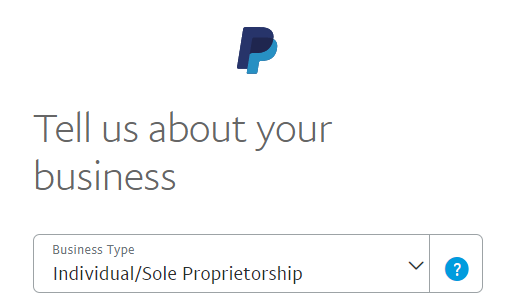
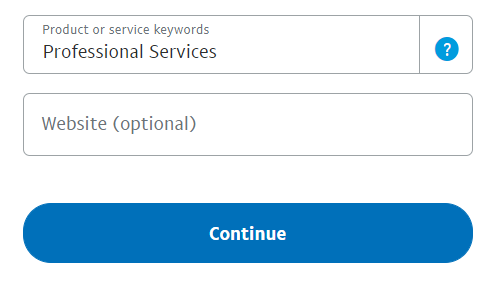
And then press: 

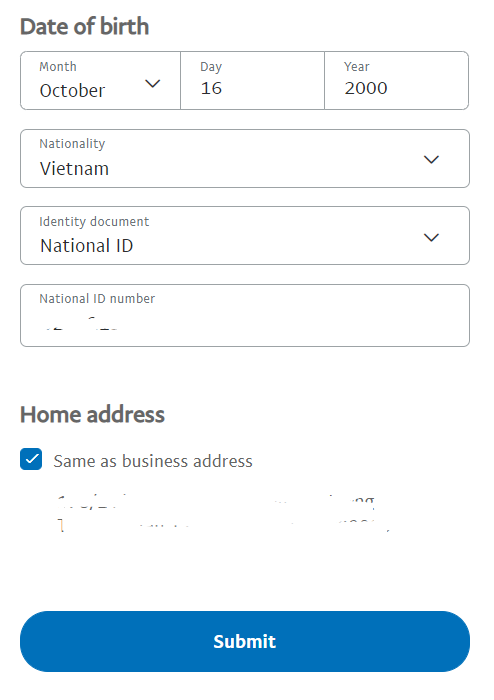
Choose Business Type: I encourage you to choose the same value in the picture.

Choose Product or service keywords: Press any letter key, it will show you the product list and then you choose any product you want.

Website: can be empty (optional).

Click Continue.



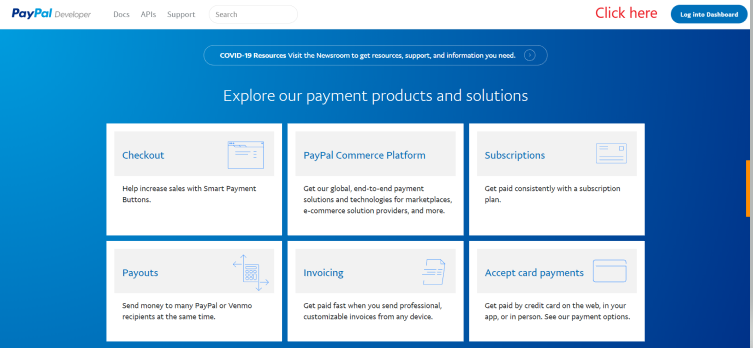
 Continue to fill in the information that they require. Check the box Same as business address to so there is no need to re-enter the address.

Press the Submit button.

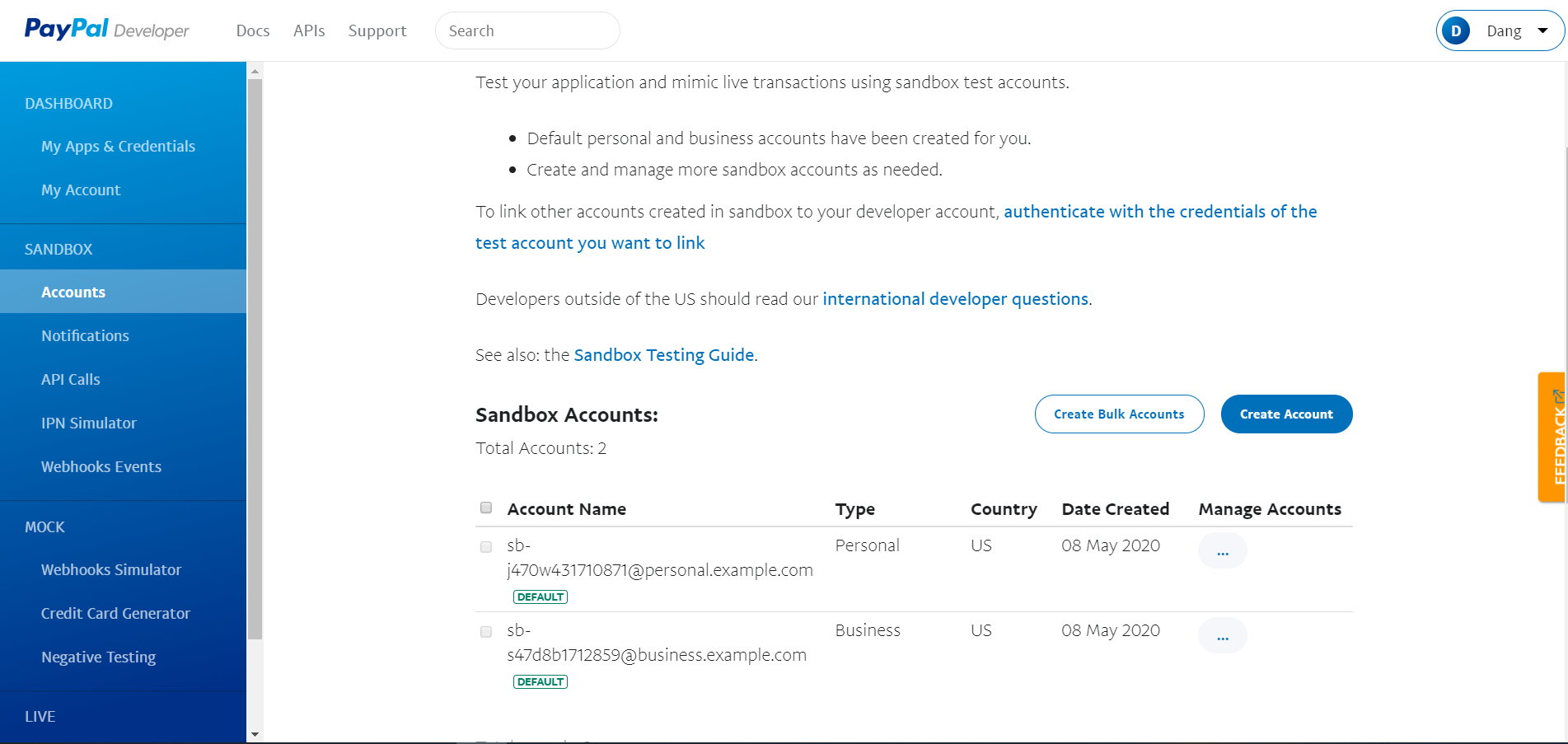
Finally, you have registered your account.

**Step 2:** Log into Developer PayPal.

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



Click Accounts in the Sandbox on the left menu.



You can see, PayPal gives you two default accounts: personal and business.

Personal account: use to pay money.

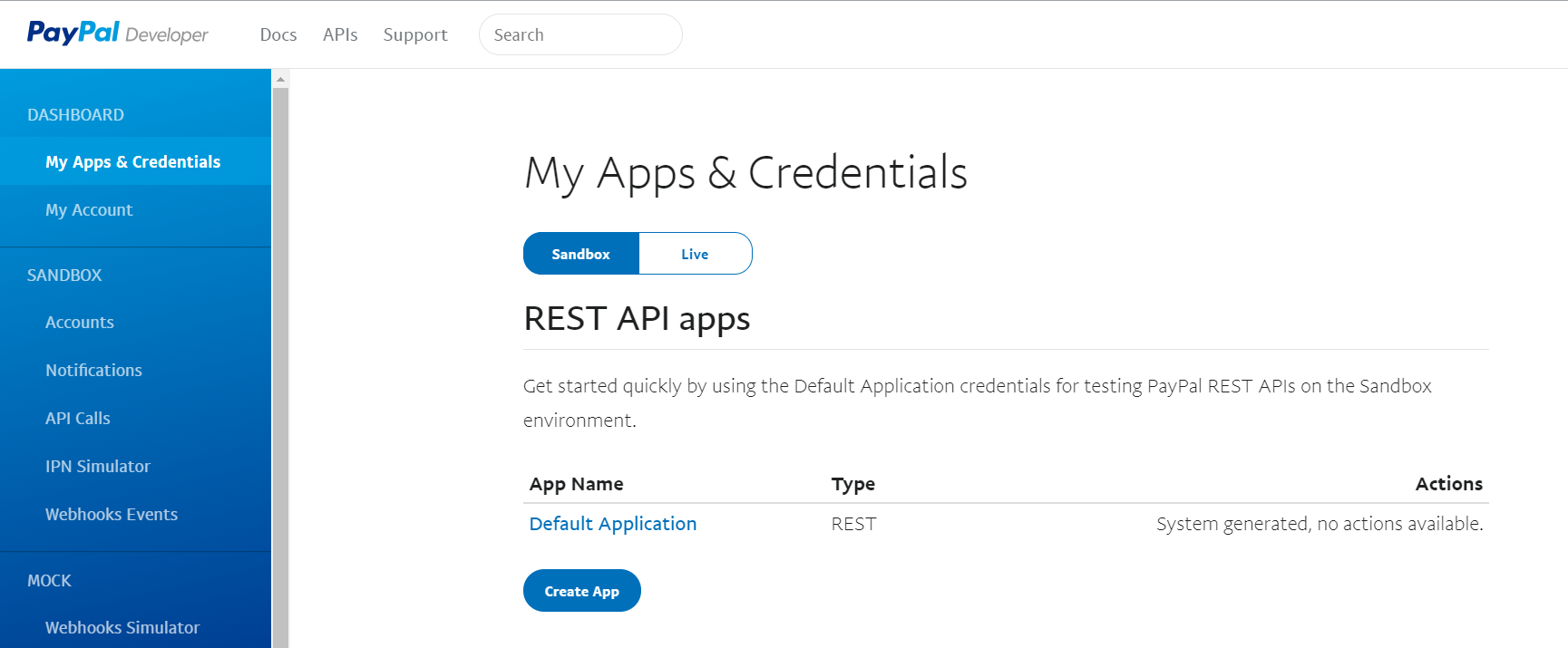
Business account: use create app and take Client ID and Secret to put in the project. When the user pays the money, that money will go into this account.

The given amount of each account is 5000$.

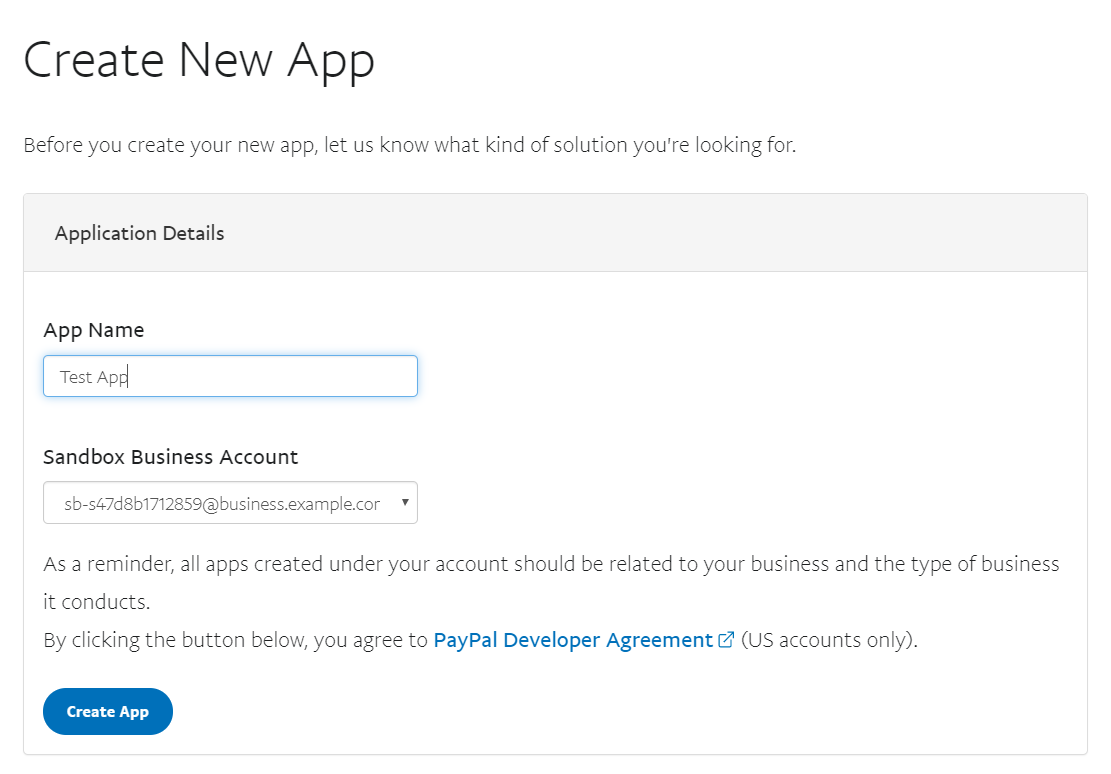
You can create account if you want, change password, view information of each account.

**Step 3:** Create App:

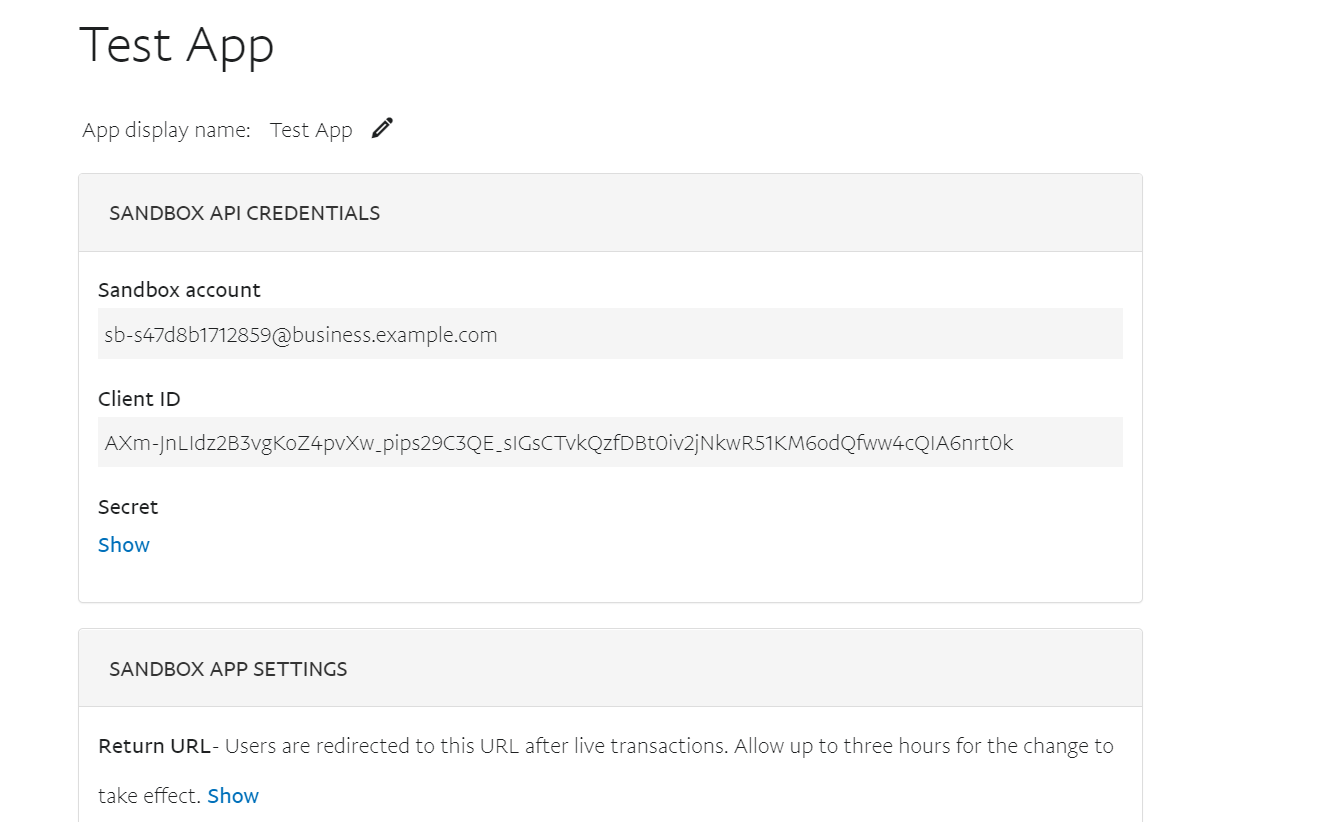
Press My Apps & Credentials. PayPal gives you available a default app.



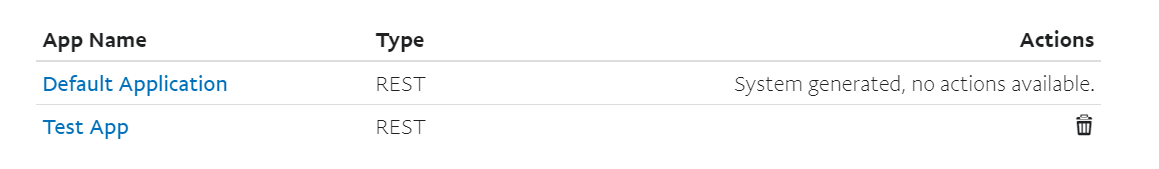
Press the Create App button to create new app.



Then press the Create App button. It shows you the necessary information about the app and you have just created. Press Show to see secret.



Back to the page with the default app, it will appear the new app you created. Just click on the app's name to display information about that app.

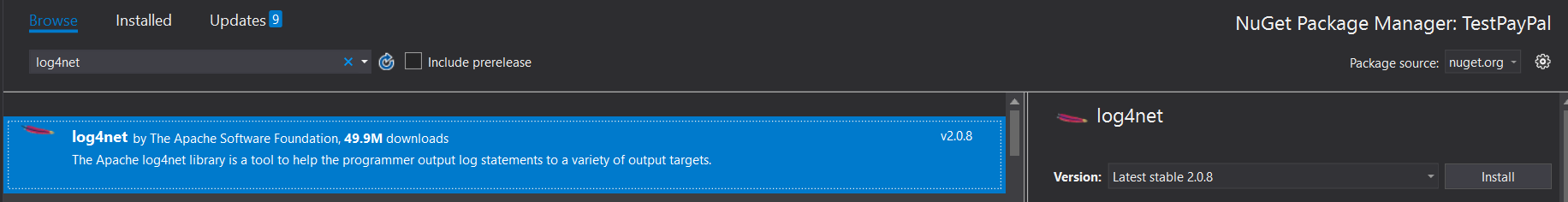


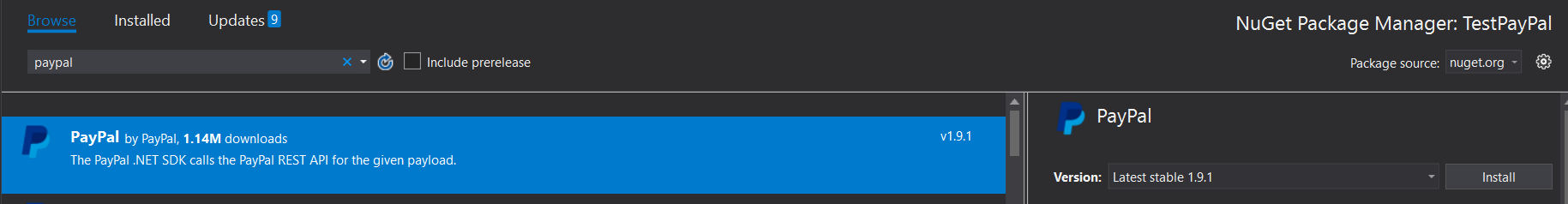
**Step 4:** Create Project API or MVC.

**Step 5:** Install Package.

Once you have created the project, right-click on the project that contains PayPal and click on Manage NuGet Packages.

Install **log4net** and **PayPal**:





**Step 6:** Edit Web.config (external).

<configSections>

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

<section name="log4net" type="log4net.Config.Log4NetConfigurationSectionHandler, log4net"/>

</configSections>

<!-- PayPal SDK settings -->

<paypal>

<settings>

<add name="mode" value="sandbox"/>

<add name="connectionTimeout" value="360000"/>

<add name="requestRetries" value="1"/>

<add name="clientId" value=""/>

<add name="clientSecret" value=""/>

</settings>

</paypal>

<!-- log4net settings -->

<log4net>

<appender name="FileAppender" type="log4net.Appender.FileAppender">

<file value="logs\log.txt"/>

<appendToFile value="true"/>

<maxSizeRollBackups value="10"/>

<maximumFileSize value="100"/>

<layout type="log4net.Layout.PatternLayout">

<header value="[Header]&#xD;&#xA;"/>

<footer value="[Header]&#xD;&#xA;"/>

<conversionPattern value="%date

[%thread] %-5level %logger [%property{NDC}] %message%newline"/>

</layout>

</appender>

<root>

<level value="DEBUG"/>

<appender-ref ref="FileAppender"/>

</root>

</log4net>

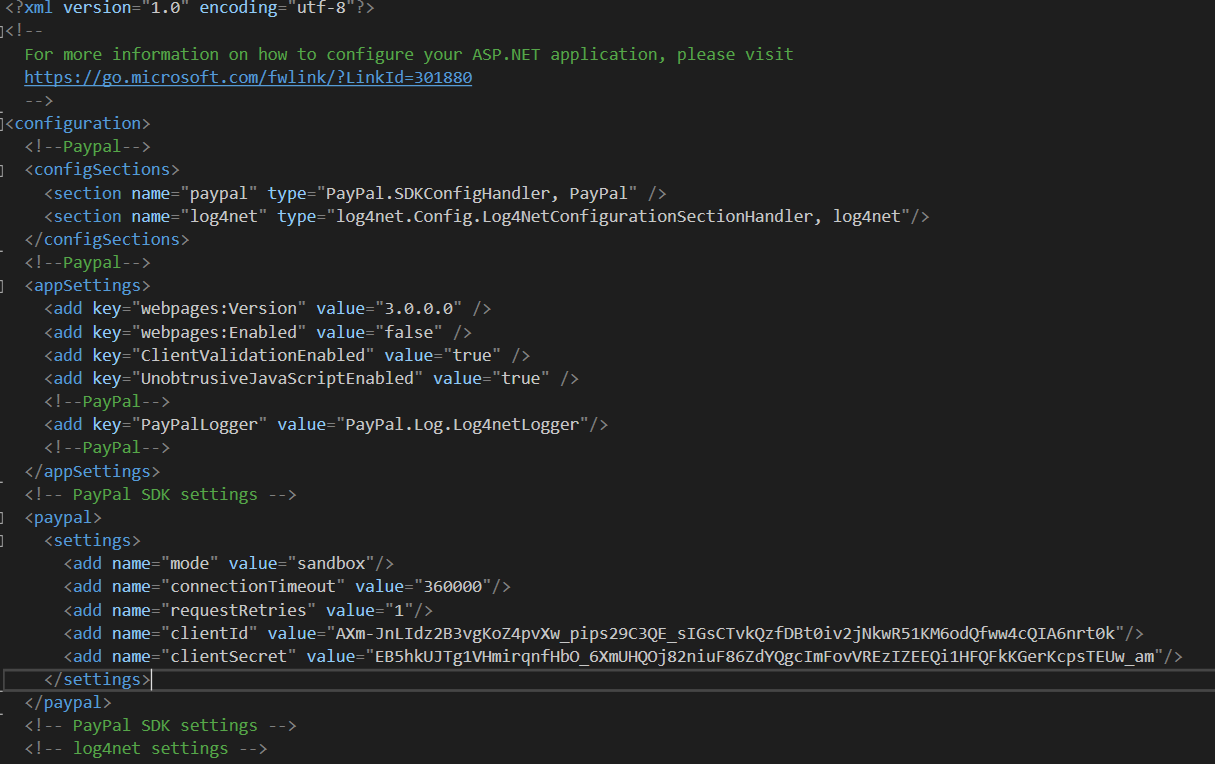
<appSettings>

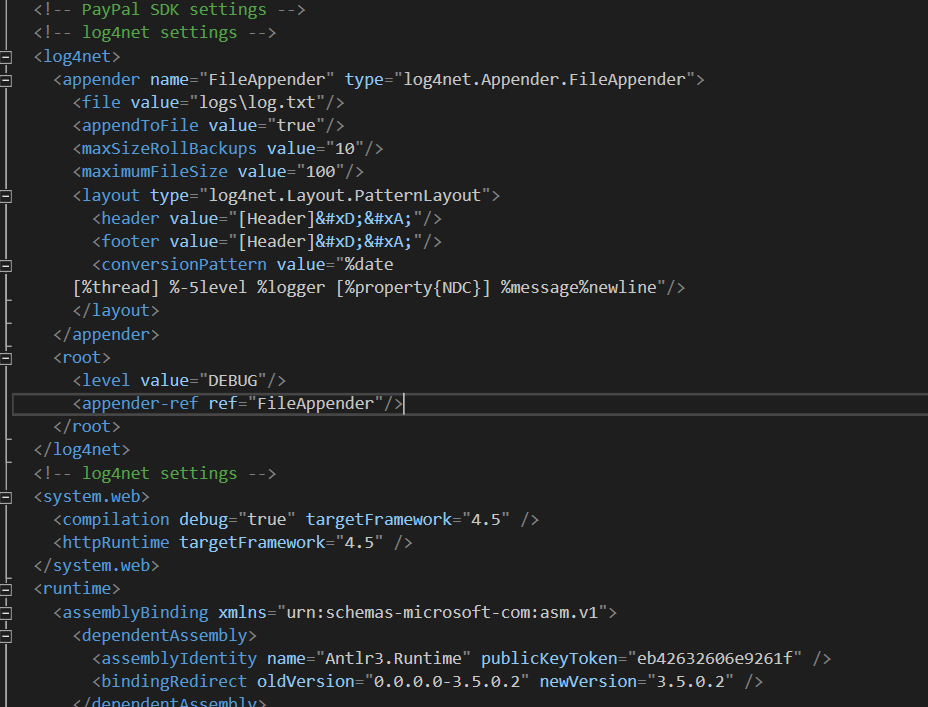
<add key="PayPalLogger" value="PayPal.Log.Log4netLogger"/>

</appSettings>

Embed this code into it accordingly.

ClientID and clientSecret, the PayPal information included in your new app.





**Step 7:** Create **PaypalConfiguration.cs** and **PaypalLogger.cs** into **Models** folder

**PaypalConfiguration.cs:** create access token to connect to PayPal.

public readonly static string ClientID;

public readonly static string ClientSecret;

static PaypalConfiguration()

{

var config = GetConfig();

ClientID = config["clientId"];

ClientSecret = config["clientSecret"];

}

public static Dictionary<string,string> GetConfig()

{

return ConfigManager.Instance.GetProperties();

}

//Create access token

private static string GetAccessToken()

{

return new OAuthTokenCredential(ClientID, ClientSecret, GetConfig()).GetAccessToken();

}

public static APIContext GetAPIContext()

{

var apiContext = new APIContext(GetAccessToken());

apiContext.Config = GetConfig();

return apiContext;

}

**PaypalLogger.cs:**

public static string LogDirectoryPath = Environment.CurrentDirectory;

public static void Log(string msg)

{

try

{

StreamWriter str = new StreamWriter(LogDirectoryPath+ "[\\PaypalError.log](file:///\\PaypalError.log)",true);

str.WriteLine("{0}--->{1}",DateTime.Now.ToString("MM/dd/yyyy HH:mm:ss"),msg);

}

catch (Exception)

{

throw;

}

}

**Step 8:** Create PayPalController (MVC 5 - Empty) into Controllers folder.

//work with paypal payment

private Payment payment;

//create a payment using an APIContext

private Payment CreatePayment(APIContext apiContext, string redirectUrl)

{

var lsItem = new ItemList() { items=new List<Item>()};

lsItem.items.Add(new Item { name = "Item 1", currency = "USD", price = "5", quantity = "1", sku = "sku" });

lsItem.items.Add(new Item { name = "Item 2", currency = "USD", price = "5", quantity = "2", sku = "sku" });

var payer = new Payer() {

payment\_method = "paypal",

payer\_info = new PayerInfo

{

email = ""

}

};

var redictUrl = new RedirectUrls() {

cancel\_url = redirectUrl,

return\_url=redirectUrl

};

var detail = new Details() { tax = "1", shipping = "1", subtotal = "15" }; //subtotal : total order, note: sum(price\*quantity) if sum is incorrect, it will have an error 400.

var amount = new Amount() { currency = "USD", details = detail, total = "17" }; //total= tax + shipping + subtotal

var transList = new List<Transaction>();

transList.Add(new Transaction {

description = "Test Payment",

invoice\_number = Convert.ToString((new Random()).Next(100000)),

amount = amount,

item\_list=lsItem,

});

this.payment = new Payment() {

intent="sale",

payer=payer,

transactions=transList,

redirect\_urls=redictUrl

};

return this.payment.Create(apiContext);

}

//create execute payment method

private Payment ExecutePayment(APIContext apiContext, string payerID, string paymentID)

{

var paymentExecute = new PaymentExecution(){payer\_id = payerID};

this.payment = new Payment() { id = paymentID};

return this.payment.Execute(apiContext, paymentExecute);

}

//create method

public ActionResult PaymentWithPaypal()

{

APIContext apiContext = PaypalConfiguration.GetAPIContext();

try

{

string payerID = Request.Params["PayerID"];

if (string.IsNullOrEmpty(payerID))

{

//create a payment

string baseUri = Request.Url.Scheme + "://" + Request.Url.Authority + "/Paypal/PaymentWithPaypal?guid=";

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

var createdPayment = CreatePayment(apiContext, baseUri + guid);

var link = createdPayment.links.GetEnumerator();

string paypalRedirectUrl = string.Empty;

while (link.MoveNext())

{

Links links = link.Current;

if (links.rel.ToLower().Trim().Equals("approval\_url"))

{

paypalRedirectUrl = links.href;

}

}

Session.Add(guid, createdPayment.id);

return Redirect(paypalRedirectUrl);

}

else {

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

var executePayment = ExecutePayment(apiContext, payerID, Session[guid] as string);

if (executePayment.state.ToLower() != "approved")

{

return View("Failure");

}

}

}

catch (PayPal.PaymentsException ex)

{

PaypalLogger.Log("Error: " + ex.Message);

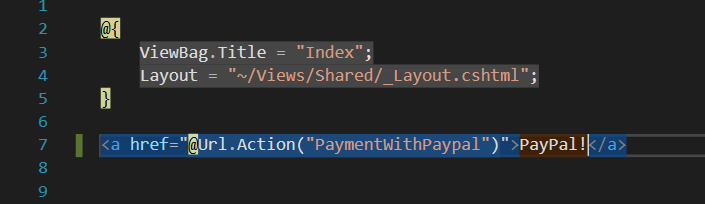
return View("Failure");

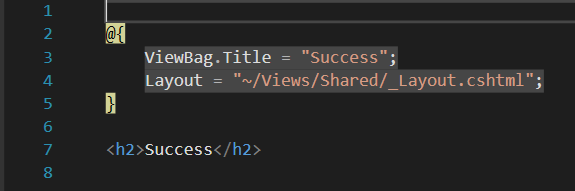
}

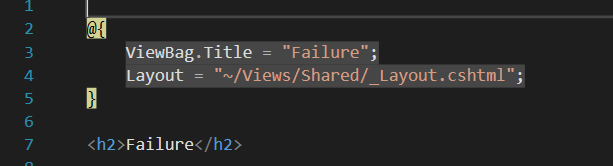
return View("Success");

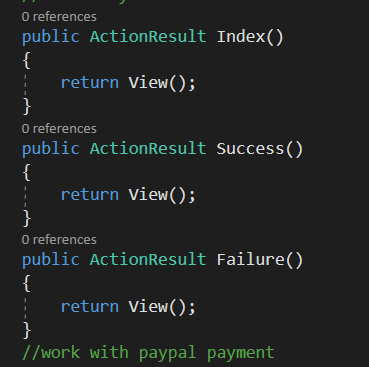
}

Create 3 Views: Index (create a tag containing Action PaymentWithPaypal to connect to PayPal), Success (return this page If payment is successful), Failure (return this page if payment is failed).

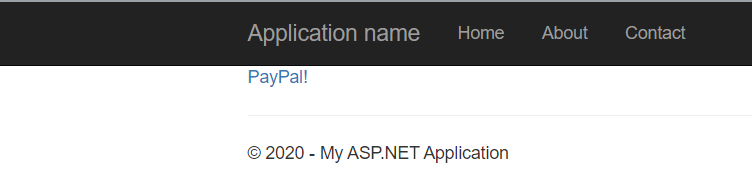








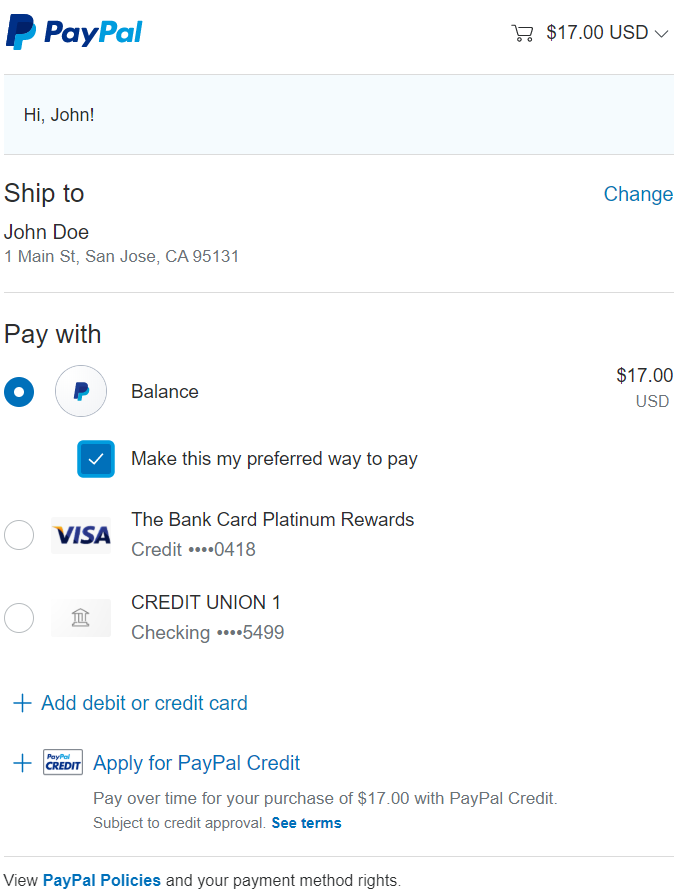
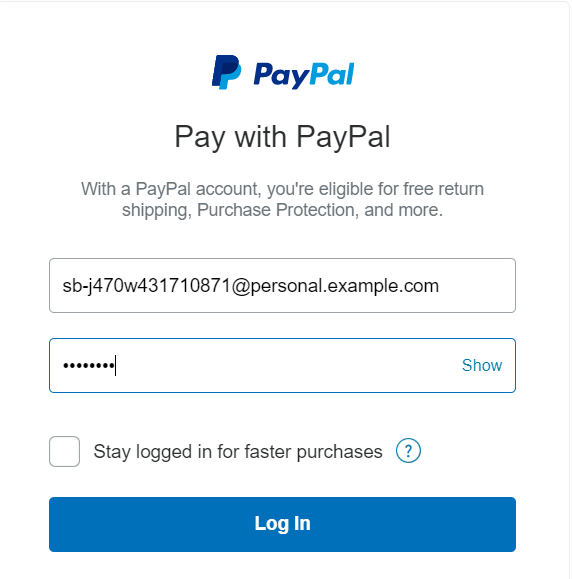
Run Index.



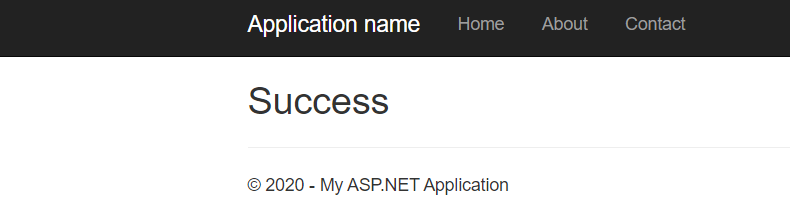
Click PayPal!

Login personal account then click Log In to pay.

🡺**Note: total of Amount should be calculated correctly otherwise PayPal will return errors.**



Click Continue to finish payment. It will return Success page if success.



*Writer: ĐTMT*

*Trainer: Mr.KB.*