Tích hợp thanh toán VNPAY cho ASP.NET

**Đăng ký tài khoản VNPay**

Ở bài viết này mình dùng môi trường Test của **VNPay**để Demo. Các bạn muốn đưa ra bản Production thì cần liên hệ với VNPay để làm hợp đồng nhé.

Còn code thì sẽ giống nhau cả thôi :D Chỉ khác cái mã và API thôi à.

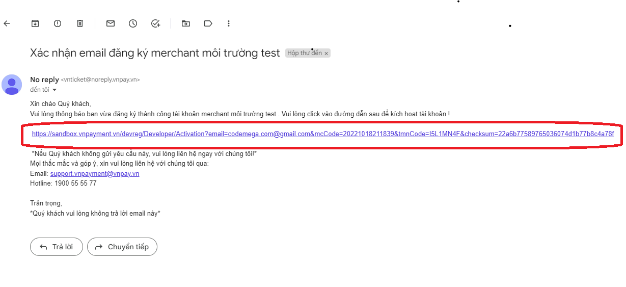
Thì đầu tiên các bạn vào link sau để đăng ký tài khoản cho môi trường Test của VNPay [TẠI ĐÂY](https://sandbox.vnpayment.vn/devreg) và điền các thông tin cần thiết vào form như hình bên dưới



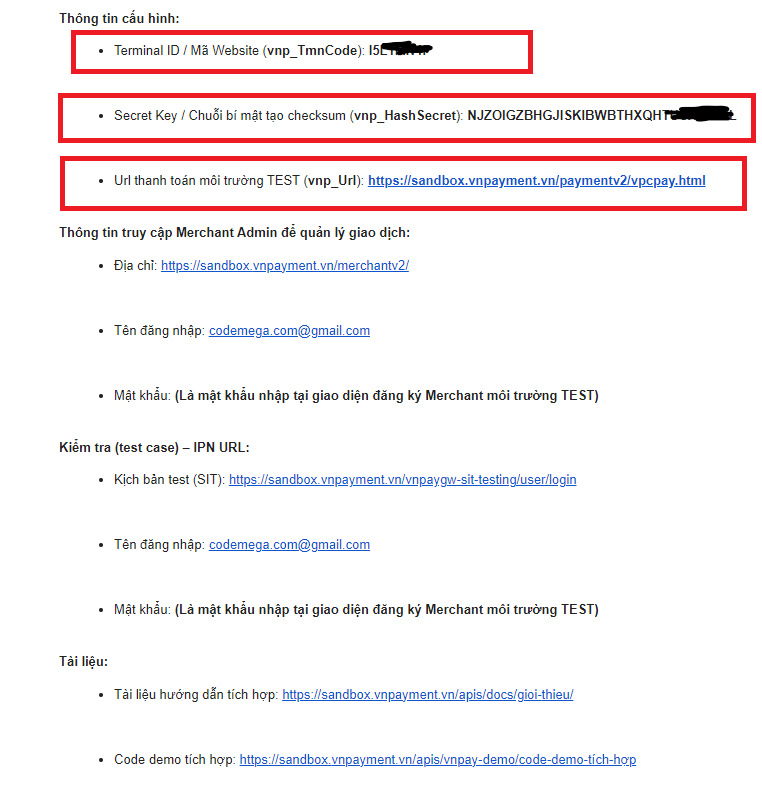
Sau khi đăng ký thành công các bạn mở Email lên và kiểm tra



Click vào link được đính kèm trong email để xác thực tài khoản



Sau khi xác thực thành công VnPay sẽ gửi một email kèm các thông tin cài đặt cho môi trường như sau:



Ở đây có một vài thông tin quan trọng mà các bạn cần lưu lại như hình trên bao gồm:

* **- vnp\_TmnCode**
* **- vnp\_HashSecret**
* **- vnp\_Url**

**Setup Code**

Bây giờ chúng ta sẽ bắt đầu vào code phần thanh toán sau khi đã đăng ký xong môi trường với VnPay

Trong file**appsetting.jso**n các bạn thêm đoạn code như sau đồng thời sửa lại các thông tin **TmnCode, HashSecret, BaseUrl**bằng thông tin của các bạn vừa đăng ký

"Vnpay": {

"TmnCode": "NJJ0R8FS", //NJJ0R8FS //**9HZKBNNN**

"HashSecret": "BYKJBHPPZKQMKBIBGGXIYKWYFAYSJXCW", //BYKJBHPPZKQMKBIBGGXIYKWYFAYSJXCW //**8HGHV2MT8QI5NLICKG28HOBLJ0AATIE6**

"BaseUrl": "https://sandbox.vnpayment.vn/paymentv2/vpcpay.html",

"Command": "pay",

"CurrCode": "VND",

"Version": "2.1.0",

"Locale": "vn",

"PaymentBackReturnUrl": "http://localhost:5172/Checkout/PaymentCallbackVnpay"

},

"TimeZoneId": "SE Asia Standard Time", // If do not us Windown OS change it to: Asia/Bangkok

Tạo 2 Model dùng để kết nối dữ liệu và dữ liệu trả về

public class PaymentInformationModel

{

public string OrderType { get; set; }

public double Amount { get; set; }

public string OrderDescription { get; set; }

public string Name { get; set; }

}

public class PaymentResponseModel

{

public string OrderDescription { get; set; }

public string TransactionId { get; set; }

public string OrderId { get; set; }

public string PaymentMethod { get; set; }

public string PaymentId { get; set; }

public bool Success { get; set; }

public string Token { get; set; }

public string VnPayResponseCode { get; set; }

}

Sau đó tạo thư viện các hàm cho cho Vnpay Library

private readonly SortedList<string, string> \_requestData = new SortedList<string, string>(new VnPayCompare());

private readonly SortedList<string, string> \_responseData = new SortedList<string, string>(new VnPayCompare());

public PaymentResponseModel GetFullResponseData(IQueryCollection collection, string hashSecret)

{

var vnPay = new VnPayLibrary();

foreach (var (key, value) in collection)

{

if (!string.IsNullOrEmpty(key) && key.StartsWith("vnp\_"))

{

vnPay.AddResponseData(key, value);

}

}

var orderId = Convert.ToInt64(vnPay.GetResponseData("vnp\_TxnRef"));

var vnPayTranId = Convert.ToInt64(vnPay.GetResponseData("vnp\_TransactionNo"));

var vnpResponseCode = vnPay.GetResponseData("vnp\_ResponseCode");

var vnpSecureHash =

collection.FirstOrDefault(k => k.Key == "vnp\_SecureHash").Value; //hash của dữ liệu trả về

var orderInfo = vnPay.GetResponseData("vnp\_OrderInfo");

var checkSignature =

vnPay.ValidateSignature(vnpSecureHash, hashSecret); //check Signature

if (!checkSignature)

return new PaymentResponseModel()

{

Success = false

};

return new PaymentResponseModel()

{

Success = true,

PaymentMethod = "VnPay",

OrderDescription = orderInfo,

OrderId = orderId.ToString(),

PaymentId = vnPayTranId.ToString(),

TransactionId = vnPayTranId.ToString(),

Token = vnpSecureHash,

VnPayResponseCode = vnpResponseCode

};

}

public string GetIpAddress(HttpContext context)

{

var ipAddress = string.Empty;

try

{

var remoteIpAddress = context.Connection.RemoteIpAddress;

if (remoteIpAddress != null)

{

if (remoteIpAddress.AddressFamily == AddressFamily.InterNetworkV6)

{

remoteIpAddress = Dns.GetHostEntry(remoteIpAddress).AddressList

.FirstOrDefault(x => x.AddressFamily == AddressFamily.InterNetwork);

}

if (remoteIpAddress != null) ipAddress = remoteIpAddress.ToString();

return ipAddress;

}

}

catch (Exception ex)

{

return ex.Message;

}

return "127.0.0.1";

}

public void AddRequestData(string key, string value)

{

if (!string.IsNullOrEmpty(value))

{

\_requestData.Add(key, value);

}

}

public void AddResponseData(string key, string value)

{

if (!string.IsNullOrEmpty(value))

{

\_responseData.Add(key, value);

}

}

public string GetResponseData(string key)

{

return \_responseData.TryGetValue(key, out var retValue) ? retValue : string.Empty;

}

public string CreateRequestUrl(string baseUrl, string vnpHashSecret)

{

var data = new StringBuilder();

foreach (var (key, value) in \_requestData.Where(kv => !string.IsNullOrEmpty(kv.Value)))

{

data.Append(WebUtility.UrlEncode(key) + "=" + WebUtility.UrlEncode(value) + "&");

}

var querystring = data.ToString();

baseUrl += "?" + querystring;

var signData = querystring;

if (signData.Length > 0)

{

signData = signData.Remove(data.Length - 1, 1);

}

var vnpSecureHash = HmacSha512(vnpHashSecret, signData);

baseUrl += "vnp\_SecureHash=" + vnpSecureHash;

return baseUrl;

}

public bool ValidateSignature(string inputHash, string secretKey)

{

var rspRaw = GetResponseData();

var myChecksum = HmacSha512(secretKey, rspRaw);

return myChecksum.Equals(inputHash, StringComparison.InvariantCultureIgnoreCase);

}

private string HmacSha512(string key, string inputData)

{

var hash = new StringBuilder();

var keyBytes = Encoding.UTF8.GetBytes(key);

var inputBytes = Encoding.UTF8.GetBytes(inputData);

using (var hmac = new HMACSHA512(keyBytes))

{

var hashValue = hmac.ComputeHash(inputBytes);

foreach (var theByte in hashValue)

{

hash.Append(theByte.ToString("x2"));

}

}

return hash.ToString();

}

private string GetResponseData()

{

var data = new StringBuilder();

if (\_responseData.ContainsKey("vnp\_SecureHashType"))

{

\_responseData.Remove("vnp\_SecureHashType");

}

if (\_responseData.ContainsKey("vnp\_SecureHash"))

{

\_responseData.Remove("vnp\_SecureHash");

}

foreach (var (key, value) in \_responseData.Where(kv => !string.IsNullOrEmpty(kv.Value)))

{

data.Append(WebUtility.UrlEncode(key) + "=" + WebUtility.UrlEncode(value) + "&");

}

//remove last '&'

if (data.Length > 0)

{

data.Remove(data.Length - 1, 1);

}

return data.ToString();

}

}

public class VnPayCompare : IComparer<string>

{

public int Compare(string x, string y)

{

if (x == y) return 0;

if (x == null) return -1;

if (y == null) return 1;

var vnpCompare = CompareInfo.GetCompareInfo("en-US");

return vnpCompare.Compare(x, y, CompareOptions.Ordinal);

}

}

Mình sẽ tạo ra 1 interfaces có tên **IVnPayService**và 1 class **VnPayService**sẽ implement interfaces **IVnPayService**

**IVnPayService**sẽ có 2 method như sau

string CreatePaymentUrl(PaymentInformationModel model, HttpContext context);

PaymentResponseModel PaymentExecute(IQueryCollection collections);

**CreatePaymentUrl**sẽ có trách nhiệm tạo ra URL thanh toán tại VnPay. Còn **PaymentExecute**sẽ thực kiểm tra thông tin giao dịch và sẽ lưu lại thông tin đó sau khi thanh toán thành công.

**CreatePaymentUrl**nhận vào một object có tên **PaymentInformationModel** model này này sẽ chứa các thông tin của hóa đơn thanh toán. Và một **HttpContext**để lấy địa chỉ IP Address của client thanh toán đơn hàng đó.

**VnPayService**sẽ có 2 method như sau

 private readonly IConfiguration \_configuration;

        public VnPayService( IConfiguration configuration)

        {

            \_configuration = configuration;

        }

public string CreatePaymentUrl(PaymentInformationModel model, HttpContext context)

        {

            var timeZoneById = TimeZoneInfo.FindSystemTimeZoneById(\_configuration["TimeZoneId"]);

            var timeNow = TimeZoneInfo.ConvertTimeFromUtc(DateTime.UtcNow, timeZoneById);

            var tick = DateTime.Now.Ticks.ToString();

            var pay = new VnPayLibrary();

            var urlCallBack = \_configuration["PaymentCallBack:ReturnUrl"];

            pay.AddRequestData("vnp\_Version", \_configuration["Vnpay:Version"]);

            pay.AddRequestData("vnp\_Command", \_configuration["Vnpay:Command"]);

            pay.AddRequestData("vnp\_TmnCode", \_configuration["Vnpay:TmnCode"]);

            pay.AddRequestData("vnp\_Amount", ((int)model.Amount \* 100).ToString());

            pay.AddRequestData("vnp\_CreateDate", timeNow.ToString("yyyyMMddHHmmss"));

            pay.AddRequestData("vnp\_CurrCode", \_configuration["Vnpay:CurrCode"]);

            pay.AddRequestData("vnp\_IpAddr", pay.GetIpAddress(context));

            pay.AddRequestData("vnp\_Locale", \_configuration["Vnpay:Locale"]);

            pay.AddRequestData("vnp\_OrderInfo", $"{model.Name} {model.OrderDescription} {model.Amount}");

            pay.AddRequestData("vnp\_OrderType", model.OrderType);

            pay.AddRequestData("vnp\_ReturnUrl", urlCallBack);

            pay.AddRequestData("vnp\_TxnRef", tick);

            var paymentUrl =

                pay.CreateRequestUrl(\_configuration["Vnpay:BaseUrl"], \_configuration["Vnpay:HashSecret"]);

            return paymentUrl;

        }

public PaymentResponseModel PaymentExecute(IQueryCollection collections)

        {

            var pay = new VnPayLibrary();

            var response = pay.GetFullResponseData(collections, \_configuration["Vnpay:HashSecret"]);

            return response;

        }

Đăng ký program.cs

//Connect VNPay API

builder.Services.AddScoped<IVnPayService, VnPayService>();

Hàm trong Controller Payment

public class PaymentController : Controller

{

private readonly IVnPayService \_vnPayService;

public PaymentController(IVnPayService vnPayService)

{

\_vnPayService = vnPayService;

}

public IActionResult CreatePaymentUrlVnpay(PaymentInformationModel model)

{

var url = \_vnPayService.CreatePaymentUrl(model, HttpContext);

return Redirect(url);

}

[HttpGet]

public IActionResult PaymentCallbackVnpay()

{

var response = \_vnPayService.PaymentExecute(Request.Query);

return Json(response);

}

}

Thông tin tài khoản test Vnpay

[Demo Cổng thanh toán VNPAY · Cổng thanh toán VNPAY](https://sandbox.vnpayment.vn/apis/vnpay-demo/)