

# TẠO THANH TOÁN BẰNG API

Tác giả : PA

## 1. Xem Video Hướng Dẫn


<https://www.youtube.com/watch?v=GEnt7jmlUiI>

## 2. Đăng Ký Merchant Môi Trường Test


- Tên Website và URL có thể đặt tùy ý vì không quan trọng. Sẽ được gửi mail thông báo kích hoạt và mail hướng dẫn các thông tin cần thiết




### ĐĂNG KÝ MERCHANT MÔI TRƯỜNG TEST




Ecommerce Website




www.magicshop.com.vn



ioephuocan@gmail.com




Mật khẩu



Nhập lại mật khẩu

Nhập mã xác nhận

9VVAYT



Đăng ký

## Thông tin tích hợp cho Merchant Cổng TT VNPAY



No reply <no-reply@noreply.vnpay.vn>  
to me

9:07 PM (4 minutes ago) ☆ 😊 ↩ ⋮

Kính gửi anh/chị,  
VNPAY xin gửi anh/chị thông tin kết nối vào môi trường test của Cổng thanh toán VNPAY.  
Xin lưu ý:

- Thông tin dưới đây là môi trường Sandbox của VNPAY, sử dụng để kết nối kiểm thử hệ thống. Merchant không sử dụng thông tin này để đưa ra cho khách hàng thanh toán thật.
- Merchant cần tạo địa chỉ IPN (server call server) sử dụng cập nhật tình trạng thanh toán (trạng thái thanh toán) cho giao dịch. Merchant cần gửi cho VNPAY URL này.

### Thông tin cấu hình:

- Terminal ID / Mã Website (vnp\_TmnCode): HS7H0S37
- Secret Key / Chuỗi bí mật tạo checksum (vnp\_HashSecret): SRXWHXLMCBGSRMNOBIXMODZSMHGXH
- Url thanh toán môi trường TEST (vnp\_Url): <https://sandbox.vnpayment.vn/paymentv2/vpcpay.html>

### 3. Tạo thư mục Helpers và tạo lớp VnPayLibrary

- Copy nội dung của VnPayLibrary trong file sau :

[https://docs.google.com/document/d/1XaOZJRJCyt0\\_WcHsxXi1xrzcC\\_8iKIUzA\\_xhk\\_V40j8/edit](https://docs.google.com/document/d/1XaOZJRJCyt0_WcHsxXi1xrzcC_8iKIUzA_xhk_V40j8/edit)

### 4. Tạo lớp VnPayResponseModel

- Copy nội dung của VnPayResponseModel cũng trong file trên

Và thêm phần dưới đây vào

```
public class VnPaymentRequestModel
{
    public int OrderId { get; set; }
    public double Amount { get; set; }
    public DateTime CreatedDate { get; set; }
}
```

### 5. Định nghĩa các hằng số trong appsettings.json

```
"VnPay": {
  "TmnCode": "HS7H0S37",
  "HashSecret": "SRXWHXLMCBGSRMNOBIXMODZSMHGXH",
  "BaseUrl": "https://sandbox.vnpayment.vn/paymentv2/vpcpay.html",
  "Version": "2.1.0",
  "Command": "pay",
  "CurrCode": "VND",
  "Locale": "vn",
  "PaymentBackReturnUrl": "https://localhost:44322/Cart/PaymentCallBack"
},
```

- Các thông tin về TmnCode, HashSecret và BaseUrl thì dựa theo mail được gửi khi đăng ký Merchant Test

## 6. Tạo Interface IvnPayService

```
public interface IVnPayService
{
    string CreatePaymentUrl(HttpContext context, VnPaymentRequestModel model);
    VnPaymentResponseModel PaymentExecute(IQueryCollection collections);
}
```

## 7. Tạo lớp VnPayService

```
public class VnPayService : IVnPayService
{
    private readonly IConfiguration _config;
    public VnPayService(IConfiguration config)
    {
        _config = config;
    }

    public string CreatePaymentUrl(HttpContext context, VnPaymentRequestModel
model)
    {
        var tick = DateTime.Now.Ticks.ToString();

        var vnpay = new VnPayLibrary();
        vnpay.AddRequestData("vnp_Version", _config["VnPay:Version"]);
        vnpay.AddRequestData("vnp_Command", _config["VnPay:Command"]);
        vnpay.AddRequestData("vnp_TmnCode", _config["VnPay:TmnCode"]);
        vnpay.AddRequestData("vnp_Amount", (model.Amount * 100).ToString());

        vnpay.AddRequestData("vnp_CreateDate",
model.CreatedDate.ToString("yyyyMMddHHmmss"));
        vnpay.AddRequestData("vnp_CurrCode", _config["VnPay:CurrCode"]);
        vnpay.AddRequestData("vnp_IpAddr", Utils.GetIpAddress(context));
        vnpay.AddRequestData("vnp_Locale", _config["VnPay:Locale"]);

        vnpay.AddRequestData("vnp_OrderInfo", "Thanh toán đơn hàng:" +
model.OrderId);
        vnpay.AddRequestData("vnp_OrderType", "other");
        vnpay.AddRequestData("vnp_ReturnUrl",
_config["VnPay:PaymentBackReturnUrl"]);
        vnpay.AddRequestData("vnp_TxnRef", tick);

        var paymentUrl = vnpay.CreateRequestUrl(_config["VnPay:BaseUrl"],
_config["VnPay:HashSecret"]);

        return paymentUrl;
    }

    public VnPaymentResponseModel PaymentExecute(IQueryCollection collections)
    {
        var vnpay = new VnPayLibrary();
        foreach (var (key, value) in collections)
```

```

        {
            if (!string.IsNullOrEmpty(key) && key.StartsWith("vnp_"))
            {
                vnnpay.AddResponseData(key, value.ToString());
            }
        }

        var vnp_orderId = Convert.ToInt64(vnnpay.GetResponseData("vnp_TxnRef"));
        var vnp_TransactionId =
        Convert.ToInt64(vnnpay.GetResponseData("vnp_TransactionNo"));
        var vnp_SecureHash = collections.FirstOrDefault(p => p.Key ==
        "vnp_SecureHash").Value;
        var vnp_ResponseCode = vnnpay.GetResponseData("vnp_ResponseCode");
        var vnp_OrderInfo = vnnpay.GetResponseData("vnp_OrderInfo");

        bool checkSignature = vnnpay.ValidateSignature(vnp_SecureHash,
        _config["VnPay:HashSecret"]);
        if (!checkSignature)
        {
            return new VnPaymentResponseModel
            {
                Success = false
            };
        }

        return new VnPaymentResponseModel {
            Success = true,
            PaymentMethod = "VnPay",
            OrderDescription = vnp_OrderInfo,
            OrderId = vnp_orderId.ToString(),
            TransactionId = vnp_TransactionId.ToString(),
            Token = vnp_SecureHash,
            VnPayResponseCode = vnp_ResponseCode,
        };
    }
}

```

## 8. Chỉnh sửa view Checkout để điều hướng tới VnPay ( Đã hoàn thiện trước đó )

## 9. Chỉnh sửa CartController

### a. Chỉnh Phương Thức ProcessCheckoutVnPay

```

[HttpPost]
[ValidateAntiForgeryToken]
public async Task<ActionResult> ProcessCheckoutVNPAY()
{
    var user = User.FindFirst(ClaimTypes.NameIdentifier);
    if (user == null)
    {
        return RedirectToAction("Login", "Account");
    }
    string userId = user.Value;
}

```

```

        var listCart = await _DbContext.CartItems
            .Include(ci => ci.Product)
            .Where(ci => ci.UserId == userId)
            .ToListAsync();

        double total = 0;
        foreach (var cartItem in listCart)
            total += cartItem.Product.Price * cartItem.Quantity;

        var vnPayModel = new VnPaymentRequestModel
        {
            Amount = total,
            CreatedDate = DateTime.Now,
            OrderId = 1,
        };

        return Redirect(_vnPayService.CreatePaymentUrl(HttpContext, vnPayModel));
    }

```

## b. Thêm Phương Thức PaymentCallback

```

public async Task<IActionResult> PaymentCallBack()
{
    var response = _vnPayService.PaymentExecute(Request.Query);

    if (response == null || response.VnPayResponseCode != "00")
    {
        return RedirectToAction("ShoppingCart");
    }

    try
    {
        var user = User.FindFirst(ClaimTypes.NameIdentifier);
        if (user == null)
        {
            return RedirectToAction("Login", "Account");
        }
        string userId = user.Value;

        var listCart = await _DbContext.CartItems
            .Include(ci => ci.Product)
            .Where(ci => ci.UserId == userId)
            .ToListAsync();

        double total = 0;
        foreach (var cartItem in listCart)
            total += cartItem.Product.Price * cartItem.Quantity;

        var order = new Order
        {
            UserId = userId,
            Status = DeliveryStatus.Processing.ToString(),
            Date = DateTime.Now,
            ShippingCost = 0,
            Total = total,
            Paid = true,
            PayMethod = PayMethod.VNPay.ToString(),
            OrderDetails = new List<OrderDetail>()

```

```

};

await _DbContext.Orders.AddAsync(order);
await _DbContext.SaveChangesAsync();

foreach (var cartItem in listCart)
{
    var orderDetail = new OrderDetail
    {
        OrderId = order.Id,
        ProductId = cartItem.ProductId,
        Quantity = cartItem.Quantity,
        ProductCost = cartItem.Product.Price * cartItem.Quantity,
        UnitPrice = cartItem.Product.Price
    };
    await _DbContext.OrderDetails.AddAsync(orderDetail);
}

_DbContext.CartItems.RemoveRange(listCart);
await _DbContext.SaveChangesAsync();

// Gửi email xác nhận đặt hàng
var email = User.FindFirstValue(ClaimTypes.Email);
var subject = "MAGIC SHOP - Xác nhận đặt hàng ( phương thức thanh toán
COD)";
var htmlMessage =
    "<p>Xin chào!,</p>\r\n    " +
    "<p>Cảm ơn bạn đã đặt hàng tại cửa hàng của chúng tôi. Đơn hàng
của bạn đã được nhận và đang được xử lý.</p>" +
    "<p>Chúng tôi hy vọng bạn sẽ hài lòng về sản phẩm.</p>" +
    "<p>Chúc bạn một ngày tốt lành♥☐</p>" +
    "<br>" +
    "<p>Thân mến," +
    "<p>Magic Shop</p>";
await _sendMailService.SendEmailAsync(email, subject, htmlMessage);

return RedirectToAction("Profile", "User");
}
catch
{
    return RedirectToAction("ShoppingCart");
}
}

```

## 10. Kiểm thử kết quả

Dùng các api thông tin thẻ được cung cấp ở trang sau :

<https://sandbox.vnpayment.vn/apis/vnpay-demo/>

→ Đây là thông tin dùng để thực hiện thanh toán thành công với VnPay (Copy qua để test )

Số Thẻ : 9704198526191432198

Tên Chủ Thẻ : NGUYEN VAN A

Ngày Phát Hành : 07/15

OTP : 123456


## DEMO

+ Chọn thanh toán VNPAY


### Checkout

Trang chủ > Thanh toán

#### TÓM TẮT ĐƠN HÀNG



Dây chuyền Bạc kim PNJ 0000W060001  
3 x 4.150.000 đ



Nhẫn Vàng 18K đính đá ECZ PNJ XMXY010580  
1 x 9.127.000 đ


Tổng cộng: 21.577.000 đ

Shipping: Free

**Thành tiền 21.577.000 đ**

#### PHƯƠNG THỨC THANH TOÁN

Thanh toán khi nhận hàng



**THANH TOÁN NGAY**

+ Chọn thanh toán bằng thẻ nội địa và tài khoản ngân hàng



### Chọn phương thức thanh toán

Ứng dụng thanh toán hỗ trợ **VNPAY<sup>QR</sup>**



Thẻ nội địa và tài khoản ngân hàng



+ Chọn ngân hàng **NCB**

Thẻ nội địa và tài khoản ngân hàng



🔍 Tìm kiếm...

 **Vietcombank**

 **VietinBank**

 **BIDV**

 **AGRIBANK**

 **Sacombank**

 **MB**

 **TECHCOMBANK**

 **ACB**

 **VPBank**

 **DONGA Bank**

 **SHB**

 **EXIMBANK**

 **HSBC**


 **TPBank**

 **NCB**  
NATIONAL CITIZEN BANK

 **OCEAN BANK**



+ Nhập các thông tin được cung cấp ở trên hoặc theo đường link nếu muốn test các trường hợp



Giao dịch hết hạn sau **11** : **39**

Thông tin đơn hàng

Số tiền thanh toán  
**21.577.000**<sup>VND</sup>

Giá trị đơn hàng  
**21.577.000**<sup>VND</sup>


Phí giao dịch  
**0**<sup>VND</sup>

Mã đơn hàng  
**638464939229043440**

Nhà cung cấp  
**Công ty CTT HTT1**

Thanh toán qua Ngân hàng NCB


Thẻ nội địa


Số thẻ  


Tên chủ thẻ

Ngày phát hành ⓘ

Mã khuyến mại

 Chọn hoặc nhập mã

 [Điều kiện sử dụng dịch vụ](#)

Hủy thanh toán

Tiếp tục

+ Nhập mã OTP



Giao dịch hết hạn sau **14 : 23**

### Thông tin đơn hàng

Số tiền thanh toán  
**9.127.000<sup>VND</sup>**

Giá trị đơn hàng  
**9.127.000<sup>VND</sup>**

Phí giao dịch  
0 VND

Mã đơn hàng  
638464930171429388

Nhà cung cấp  
Công ty CTT HTT1

## Xác thực OTP



Nhập mã OTP



OTP đã được gửi về số điện thoại đăng ký. Quý khách vui lòng nhập OTP để tiếp tục thực hiện giao dịch.

Hủy

## Thanh toán



1900.5555.77



[hotrovnpay.vn@vnpay.vn](mailto:hotrovnpay.vn@vnpay.vn)



secure  
GlobalSign  
CERTIFICATES



+ Đơn hàng được lưu và thanh toán thành công

100 %

Results Messages

	Id	Total	ShippingCost	UserId	Status	Date	Paid	PayMethod
1	1	7000000	0	4bf8bdc-c1f2-4f86-965f-2319c0783adf	Processing	2024-03-18 18:26:31.0737943	0	COD
2	2	19400000	0	4bf8bdc-c1f2-4f86-965f-2319c0783adf	Processing	2024-03-19 20:46:44.3795551	0	COD
3	3	35000000	0	4bf8bdc-c1f2-4f86-965f-2319c0783adf	Processing	2024-03-19 20:53:35.2627714	0	COD
4	4	4949900	0	4bf8bdc-c1f2-4f86-965f-2319c0783adf	Processing	2024-03-20 00:47:31.9760634	0	COD
5	5	14076900	0	4bf8bdc-c1f2-4f86-965f-2319c0783adf	Processing	2024-03-20 00:51:07.5452792	0	VNPay
6	6	9127000	0	4bf8bdc-c1f2-4f86-965f-2319c0783adf	Processing	2024-03-20 00:55:47.4525038	1	VNPay