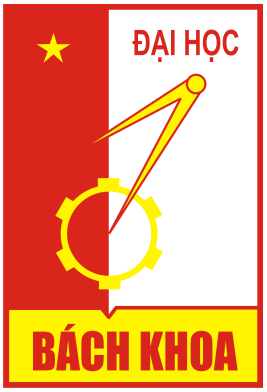
Web Information System Project

About How to Use VnPay to Transfer Money

Nguyen Ngoc Lam - 20162316

A report presented what I have gained
from doing this project



School of Information and Communication
Hanoi University of Science and Technology
Friday 19th June, 2020

Contents

1 Overview about payment gateway

1.1 Payment gateway

1.1.1 Definition

A payment gateway is a merchant service[1] provided by an e-commerce application service provider that authorizes credit card or direct payments processing for e-businesses, online retailers, bricks and clicks[2], or traditional brick and mortar[3].
The payment gateway may be provided by a bank to its customers, but can be provided by a specialized financial service provider as a separate service, such as a payment service provider.
A payment gateway facilitates a payment transaction by the transfer of information between a payment portal (such as a website, mobile phone or interactive voice response service) and the front end processor or acquiring bank[4] Some payment gateways offer white label services, which allow payment service providers, e-commerce platforms, ISOs, resellers, or acquiring banks to fully brand the payment gateway's technology as their own. This means PSPs[5] or other third parties can own the end-to-end user experience without bringing payments operations, and additional risk management and compliance responsibility, in house

1.1.2 Typical process

1. Customer places an order on website.

2. Encrypts the information to be sent between the user device and the merchant's web server.

3. The merchant then forwards the transaction details to their payment gateway. This is another encrypted connection to the payment server hosted by the payment gateway. Some gateways allows users to bypass the merchant's web server and sent the information straight to them.

[1] refers to merchant processing services that enables a business to accept a transaction payment through a secure (encrypted) channel using the customer's credit card or debit card

[2] a business model by which a company integrates both offline (bricks) and online (clicks) presences

[3] refers to a physical presence of an organization or business in a building or other structure

[4] is a bank or financial institution that processes credit or debit card payments on behalf of a merchant like Visa or Master Card

[5] Payment Service Provider

4. The payment gateway converts the message from XML to a message format[6] that understood by EFT Switches and then forwards the transaction information to the payment processor used by the merchant's acquiring bank.

5. The payment processor forwards the transaction information to the card association[7]

6. The credit card issuing bank receives the authorization request, verifies the credit or debit available and then sends a response back to the processor (via the same process as the request for authorization) with a response code.

7. The processor forwards the authorization response to the payment gateway.

8. The payment gateway receives the response, and forwards it onto the interface that was used to process the payment, where it is interpreted as a relevant response, then relayed back to the merchant and cardholder. This is known as the Authorization or "Auth".

9. The merchant then fulfills the order and the above process can be repeated but this time to "Clear" the authorization by consummating the transaction.

10. The merchant submits all their approved authorizations, in a "batch" (end of the day), to their acquiring bank for settlement via its processor to reduced the number of "Clear".

11. The acquiring bank makes the batch settlement request of the credit card issuer who makes a settlement payment to the acquiring bank (the next day in most cases).

12. The acquiring bank subsequently deposits the total of the approved funds into the merchant's nominated account (the same day or next day).

1.1.3 Advantages over other payment methods

- The merchant does not need to contact directly to bank thus can save time and human resource for them.

[6] like ISO 8583

[7] is a network of issuing banks and acquiring banks that process payment cards of a specific brand

- Concentrating all the online transactions to a few gateway instead of each merchant have to create their own gateways.
- Safer than COD[8] model for the merchants as they can ship the goods after the bank completed their verification.
- Less exchange of cash lead to faster and easier process for users since most of jobs was done behind a "closed door", where both customers and merchants cannot see.
- On a more recent note, reduced the human interactions thus preventing diseases from spreading.

1.1.4 Disadvantages compared to other payment methods

- Having an intermediate between you and the bank was never a good idea as they can see your information. If there was a security breach in the Payment gateway, your credit card information could be stolen.
- It is slower than a perfect COD scenario where the merchant can trust the customer to pay every time they order.
- Two encrypted connections from the firs few steps can be a target for hackers.
- The issue of credit card frauds is not addressed in the process.

1.2 VnPay

1.2.1 Overview

- It is a payment gateway connecting merchants with banks.
- It allows customers to pay using credit/debit card, QR code and mobile banking applications on the phones.

1.2.2 Benefits of using VnPay inside Vietnam

- It is one of the most widely used payment medium in Vietnam.
- It has been incorporated into many mobile applications of popular banks like Vietcombank, BIDV and Vietinbank

[8]Cash On Delivery

- It has also been used in many e-wallets like viettelpay or VinID
- VnPayQR is supported by a lot of vendors in Vietnam
- It is easy to incorporated into your website by using code or by open-sourced plugins

1.2.3 How to incorporated

You can incorporated VnPay by 3 ways:

1. Through code: It delivered the most completed package but vendors need to have someone who can understand the demo code and how it works to deploy it.
2. Through Third party open-sourced software: Vnpay has modules, libraries on third party open-sourced software like Magento, OpenCart and PrestaShop
3. Through embedded code: Simple.

1.2.4 Deployment process

1. Preparation
2. Incorporated in test environment
3. Testing
4. Go live

1.2.5 Fee

Applied for all VnPayQR transactions

- For transaction using domestic card or bank account, e-wallet: 0.88% of transaction value.
- For transaction using international credit/debit card: 2.2% of transaction value.

Applied for all transactions using payment gateway:

- For transaction using domestic card or bank account, e-wallet: 1.1% of transaction value with an additional fee of 1,650 Vietnam dong.
- For transaction using international credit/debit card: 2.2% of transaction value with an additional fee of 2,500 Vietnam dong

2 Learning how to use VnPay through demo application

2.1 Language supported

- PHP
- C#
- Python
- java
- NodeJS
- For this project, I will use python code.

2.2 Configuration

There are 2 configures to identify the website:

1. vnp_TmpCode: Website code on the VnPay server
2. vnp_HashSecret: Private code to authenticate (checksum) the server

2.3 Guide for incorporated

2.3.1 Create a payment URL

Payment URL contains payment details like: what type of transactions it is, what bank does the customer use, what time do they send the request, How much is the transaction.

2.3.2 Code IPN URL

IPN URl is used to get the result of processing the request on the server to the merchant using server-call-server call and the update/process the result will be done here.

2.3.3 Code ReturnURL

Vnpay sends data back to user by redirecting the browser to the website that the merchant provided. This site is used to show the result of the transaction to the end user

2.4 Demo code

References

[1] Payment gateway: https://en.wikipedia.org/wiki/Payment_gateway

[2] Merchant services: https://en.wikipedia.org/wiki/Merchant_services

[3] Bricks and clicks: https://en.wikipedia.org/wiki/Bricks_and_clicks

[4] Brick and mortar: https://en.wikipedia.org/wiki/Brick_and_mortar

[5] Acquiring bank: https://en.wikipedia.org/wiki/Acquiring_bank

[6] Card association: https://en.wikipedia.org/wiki/Card_association

[7] VnPay home page: https://vnpay.vn/

[8] Payment gateway VnPay home page: https://www.vnpayment.vn/

[9] VnPay features: https://vnpayment.vnpay.vn/tinh-nang.htm

[10] VnPay fee: https://vnpayment.vnpay.vn/phi.htm

[11] Documentation for API: https://sandbox.vnpayment.vn/apis/docs/gioi-thieu/

```python
def payment(request):
    if request.method == 'POST':
        # Process input data and build url payment
        form = PaymentForm(request.POST)
        if form.is_valid():
            order_type = form.cleaned_data['order_type']
            order_id = form.cleaned_data['order_id']
            amount = form.cleaned_data['amount']
            order_desc = form.cleaned_data['order_desc']
            bank_code = form.cleaned_data['bank_code']
            language = form.cleaned_data['language']
            ipaddr = get_client_ip(request)
            # Build URL Payment
            vnp = vnpay()
            vnp.requestData['vnp_Version'] = '2.0.0'
            vnp.requestData['vnp_Command'] = 'pay'
            vnp.requestData['vnp_TmnCode'] = settings.VNPAY_TMN_CODE
            vnp.requestData['vnp_Amount'] = amount * 100
            vnp.requestData['vnp_CurrCode'] = 'VND'
            vnp.requestData['vnp_TxnRef'] = order_id
            vnp.requestData['vnp_OrderInfo'] = order_desc
            vnp.requestData['vnp_OrderType'] = order_type
            # Check language, default: vn
            if language and language != '':
                vnp.requestData['vnp_Locale'] = language
            else:
                vnp.requestData['vnp_Locale'] = 'vn'
                # Check bank_code, if bank_code is empty, customer will be selected bank on VNPAY
            if bank_code and bank_code != "":
                vnp.requestData['vnp_BankCode'] = bank_code

            vnp.requestData['vnp_CreateDate'] = datetime.now().strftime('%Y%m%d%H%M%S')  # 20150410063022
            vnp.requestData['vnp_IpAddr'] = ipaddr
            vnp.requestData['vnp_ReturnUrl'] = settings.VNPAY_RETURN_URL
            vnpay_payment_url = vnp.get_payment_url(settings.VNPAY_PAYMENT_URL, settings.VNPAY_HASH_SECRET_KEY)
```

(a) Render Payment URL parameter

```python
            language = form.cleaned_data['language']
            ipaddr = get_client_ip(request)
            # Build URL Payment
            vnp = vnpay()
            vnp.requestData['vnp_Version'] = '2.0.0'
            vnp.requestData['vnp_Command'] = 'pay'
            vnp.requestData['vnp_TmnCode'] = settings.VNPAY_TMN_CODE
            vnp.requestData['vnp_Amount'] = amount * 100
            vnp.requestData['vnp_CurrCode'] = 'VND'
            vnp.requestData['vnp_TxnRef'] = order_id
            vnp.requestData['vnp_OrderInfo'] = order_desc
            vnp.requestData['vnp_OrderType'] = order_type
            # Check language, default: vn
            if language and language != '':
                vnp.requestData['vnp_Locale'] = language
            else:
                vnp.requestData['vnp_Locale'] = 'vn'
                # Check bank_code, if bank_code is empty, customer will be selected bank on VNPAY
            if bank_code and bank_code != "":
                vnp.requestData['vnp_BankCode'] = bank_code

            vnp.requestData['vnp_CreateDate'] = datetime.now().strftime('%Y%m%d%H%M%S')  # 20150410063022
            vnp.requestData['vnp_IpAddr'] = ipaddr
            vnp.requestData['vnp_ReturnUrl'] = settings.VNPAY_RETURN_URL
            vnpay_payment_url = vnp.get_payment_url(settings.VNPAY_PAYMENT_URL, settings.VNPAY_HASH_SECRET_KEY)
            print(vnpay_payment_url)
            if request.is_ajax():
                # Show VNPAY Popup
                result = JsonResponse({'code': '00', 'Message': 'Init Success', 'data': vnpay_payment_url})
                return result
            else:
                # Redirect to VNPAY
                return redirect(vnpay_payment_url)
        else:
            print("Form input not validate")
    else:
        return render(request, "payment.html", {"title": "Thanh toán"})
```

(b) Return the full Payment URL

Figure 1: Create Payment URL

```python
def payment_ipn(request):
    inputData = request.GET
    if inputData:
        vnp = vnpay()
        vnp.responseData = inputData.dict()
        order_id = inputData['vnp_TxnRef']
        amount = inputData['vnp_Amount']
        order_desc = inputData['vnp_OrderInfo']
        vnp_TransactionNo = inputData['vnp_TransactionNo']
        vnp_ResponseCode = inputData['vnp_ResponseCode']
        vnp_TmnCode = inputData['vnp_TmnCode']
        vnp_PayDate = inputData['vnp_PayDate']
        vnp_BankCode = inputData['vnp_BankCode']
        vnp_CardType = inputData['vnp_CardType']
        if vnp.validate_response(settings.VNPAY_HASH_SECRET_KEY):
            # Check & Update Order Status in your Database
            # Your code here
            firstTimeUpdate = True
            if firstTimeUpdate:
                if vnp_ResponseCode == '00':
                    print('Payment Success. Your code implement here')
                else:
                    print('Payment Error. Your code implement here')

                # Return VNPAY: Merchant update success
                result = JsonResponse({'RspCode': '00', 'Message': 'Confirm Success'})
            else:
                # Already Update
                result = JsonResponse({'RspCode': '02', 'Message': 'Order Already Update'})

        else:
            # Invalid Signature
            result = JsonResponse({'RspCode': '97', 'Message': 'Invalid Signature'})
    else:
        result = JsonResponse({'RspCode': '99', 'Message': 'Invalid request'})

    return result
```

Figure 2: Create IPN URL

```python
def payment_return(request):
    inputData = request.GET
    if inputData:
        vnp = vnpay()
        vnp.responseData = inputData.dict()
        order_id = inputData['vnp_TxnRef']
        amount = int(inputData['vnp_Amount']) / 100
        order_desc = inputData['vnp_OrderInfo']
        vnp_TransactionNo = inputData['vnp_TransactionNo']
        vnp_ResponseCode = inputData['vnp_ResponseCode']
        vnp_TmnCode = inputData['vnp_TmnCode']
        vnp_PayDate = inputData['vnp_PayDate']
        vnp_BankCode = inputData['vnp_BankCode']
        vnp_CardType = inputData['vnp_CardType']
        if vnp.validate_response(settings.VNPAY_HASH_SECRET_KEY):
            if vnp_ResponseCode == "00":
                return render(request, "payment_return.html", {"title": "Kết quả thanh toán",
                                                               "result": "Thành công", "order_id": order_id,
                                                               "amount": amount,
                                                               "order_desc": order_desc,
                                                               "vnp_TransactionNo": vnp_TransactionNo,
                                                               "vnp_ResponseCode": vnp_ResponseCode})
            else:
                return render(request, "payment_return.html", {"title": "Kết quả thanh toán",
                                                               "result": "Lỗi", "order_id": order_id,
                                                               "amount": amount,
                                                               "order_desc": order_desc,
                                                               "vnp_TransactionNo": vnp_TransactionNo,
                                                               "vnp_ResponseCode": vnp_ResponseCode})
        else:
            return render(request, "payment_return.html",
                          {"title": "Kết quả thanh toán", "result": "Lỗi", "order_id": order_id, "amount": amount,
                           "order_desc": order_desc, "vnp_TransactionNo": vnp_TransactionNo,
                           "vnp_ResponseCode": vnp_ResponseCode, "msg": "Sai checksum"})
    else:
        return render(request, "payment_return.html", {"title": "Kết quả thanh toán", "result": ""})
```

Figure 3: Create ReturnURL