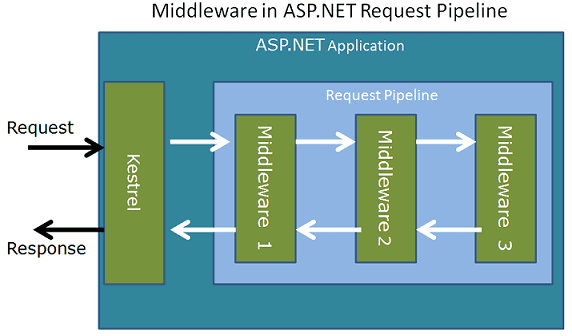
Giới thiệu về công nghệ

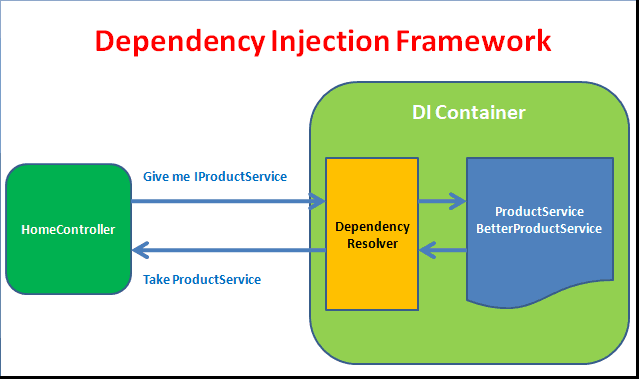
Backend

Giới thiệu về ASP.NET Core

* ASP.NET Core là một mã nguồn mở, có thể tạo ứng dụng bằng .Net core cho các nền tảng như Windows, macOS và Linux x64, x86, ARM32 và ARM64. Frameworks và APIs cung cấp cloud, IoT, client UI và machine learning.
* Một số điểm mới trong .Net core như:
  + .Net core giao diện dòng lệnh (command-line interface) là chuỗi công cụ chạy đa nền tảng hỗ trợ cho việc phát triển (developing), xây dựng dự án (building), chạy chương trình (running) và xuất bản dự án (publishing)
  + Kestrel web server: Là web server mã nguồn mở, đa nền tảng, hướng sự kiện và bất đồng bộ. Được phát triển để chạy trên bất cứ nền tảng nào. Được thêm vào mặc định trong ứng dụng .Net core
  + Middleware và Request Pipeline trong ASP.NET Core:
    - Request Pipeline: là cơ chế bắt đầu khi các request bắt đầu được xử lý với một đối tượng Request đầu vào và kết thúc với đầu ra là một response. Pipeline chỉ ra cách mà ứng dụng phản hồi với HTTP Request. Request đến từ trình duyệt đi qua pipeline và quay trở lại khi xử lý xong để trả về client. Các thành phần đơn lẻ tạo nên pipeline này được gọi là middleware.
    - Middleware: là thành phần của phần mềm đóng vai trò tác động vào request pipeline (luồng request) để xử lý chúng và tạo ra response phản hồi lại client. Mỗi một tiến trình middleware thao tác với các request nhận được từ middleware trước nó. Nó cũng có thể quyết định gọi middleware tiếp theo trong pipeline hoặc trả về response cho middleware ngay trước nó. (ngắt pipeline).

Hình 1.3: Hình ảnh Request Pipeline kết hợp Middleware

* + Cơ chế Dependency Injection trong ASP.NET Core: là một phần của ASP.NET Core. Tất cả các service của framework đều được inject khi chúng ta cần. Lợi ích Dependency Injection:
    - Dependency injection giúp thực hiện kiến trúc lỏng lẻo (loose coupling) trong phần mềm.
    - Code sẽ sạch và dễ đọc hơn
    - Tăng khả năng có thể kiểm thử và bảo trì
    - Cho phép bạn thay đổi triển khai mà không phải thay đổi quá nhiều code.

Hình 1.4: Hình ảnh mô tả cách ProductService được Inject vào HomeController

* + Vòng đời của Dependency Injection: Transient, Singleton, Scoped. Mỗi service có vòng đời khác nhau như:
    - Transient: Một thể hiện mới luôn được tạo, mỗi khi được yêu cầu.
    - Scoped: Tạo một thể hiện mới cho tất cả các scope (Tức mỗi request là một scope). Trong scope thì service được dùng lại.
    - Singleton: Service được tạo chỉ một lần duy nhất.

Ưu điểm của ASP.NET Core

* ASP.NET Core có một số thay đổi kiến trúc dẫn đến module framework nhỏ hơn.
* ASP.NET Core không còn dựa trên System.Web.dll. Nó dựa trên một tập hợp nhiều yếu tố của Nuget packages.
* Điều này cho phép tối ưu ứng dụng của mình chỉ cần những NuGet packages cần thiết.
* Lợi ích của diện tích bề mặt ứng dụng nhỏ hơn thì bảo mật chặt chẽ hơn, giảm dịch vụ, cải thiện hiệu suất và giảm chi phí.

Lí do sử dụng công nghệ:

Front-end:

Mustache Template:

Quá trình phát triển ứng dụng, chúng ta thường viết các nhiều đoạn mã HTML trong mã Javascript dưới dạng kiểu chuỗi. Gây khó khăn cho quá trình phát triển và bảo trì mã nguồn. Vì vậy thư viện MustacheJS là một khuôn mẫu HTML. Việc gắn chặt giao diện người dùng và logic khiến cho ứng dụng khó khăn hơn để phát triển và bảo trì lại sau này.

Thư viện Mustache ra đời giải quyết vấn đề này bằng cách tách các định nghĩa giao diện người dùng ra khỏi dữ liệu, Mustache giúp kết xuất giao diện nhanh hơn sau mỗi lần kết xuất sau, do cơ chế lưu cache.

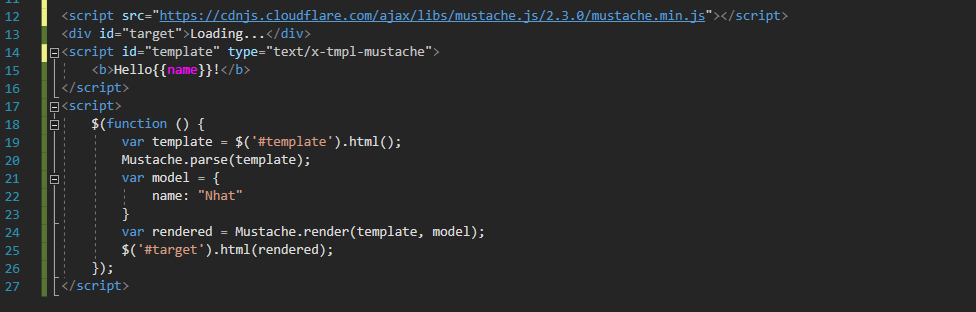
Cài đặt Mustache.js

* Cài đặt thông qua npm

Ta dùng lệnh console: npm install mustache –save.

Cách sử dụng

Một số ví dụ đơn giản

Bước 1: Việc đầu tiên cần làm là include mustache.js trong file HTML:

Bước 2: Khai báo Mustache Template sử dụng thẻ script:

* + Đặt id cho script và khai báo **type="x-tmpl-mustache"** để thông báo đây là a
  + Bên trong đoạn HTML dùng là template. Với đặc điểm quan trọng là mỗi tham số được đặt trong cặp dấu **{{variable}}.** Qua đó, Mustache sẽ biết và thay nó với đối số tương ứng khi truyền vào.

Bước 3: Sử dụng hàm html trong Jquery để xuất HTML ra giao diện

**Giới thiệu về MoMo**

**MoMo là gì?**

Là một ứng dụng Ví điện tử Momo, sản phẩm của Công ty Cổ phần Dịch vụ Di động Trực tuyến viết tắt là M\_Service. Ứng dụng cho phép bạn tạo và nạp tiền vào tài khoản MoMo để thanh toán cho hơn 200 dịch vụ như: Nạp tiền điện thoại tất cả các nhà mạng, thanh toán cách loại hoá đơn, thanh toán vay trả góp, mua sắm online, thanh toán dịch vụ ăn uống - giải trí.

Ví điện tử MoMo được cấp giấy phép và quản lý bởi Ngân hàng Nhà nước. Đồng thời là đối tác chiến lược của các**25 Ngân hàng lớn:** Vietcombank, Vietinbank, Agribank, BIDV, ACB, TPBank, VPBank, Eximbank, OCB, VIB, Shinhan Bank, SCB, Thẻ nội địa Sacombank, VRB, BAOVIET Bank, ABBANK, OceanBank, MBBank, PVcomBank, VietCapital Bank, Nam Á Bank, SHB, Bac A Bank, HDBank, SAIGONBANK.

**Ví điện tử là gì?**

Ví điện tử hay còn gọi là Ví tiền Online là một tài khoản thanh toán các giao dịch trực tuyến phổ biến nhất hiện nay như: Thanh toán tiền điện, nước, học phí, nạp tiền điện thoại, mua vé xem phim, các dịch vụ giải trí khác,...

Chức năng hoạt động của Ví điện tử thực hiện bằng cách Liên kết Tài Khoản ngân hàng, nạp tiền vào Ví và thanh toán bất kì dịch vụ có liên kết một cách đơn giản, tiện lợi.

Chỉ cần bạn có Smartphone, mạng Wifi và tài khoản đã có thể thực hiện mọi giao dịch ở mọi nơi. Bạn có thể nạp tiền nhanh chóng tại các của hàng tiện lợi, siêu thị mini, cửa hàng điện tử, điện máy như: Circle K, MiniStop, Thegioididong, Bachhoaxanh,…  
Hiện nay nhiều rất nhiều nơi chấp nhận thanh toán bằng MoMo: Tiki, Nhaccuatui, CGV, Lazada, VietjetAir, Coopmart, Lotte và một số của hàng tiện lợi như là Circle K...

**Mục đích tích hợp cổng thanh toán MoMo?**

Do trong quá trình phát triển website thì nhóm em đã tìm hiểu rất nhiều phương thức thanh toán, mục đích để có thể tạo ra sự tiện lợi nhất cho người sử dụng, quá trình nghiên cứu thị trường cổng thanh toán thì dịch vụ MoMo là cổng thanh toán được ưa chuộng nhất Việt Nam hiện nay và được đông đảo người sử dụng, để đáp ứng được nhu cầu của đề tài cho nên nhóm em chọn cổng thanh toán MoMo để tích hợp vào website, chỉ với vài cú đúp chuột là có thể mua được khoá học một cách nhanh chóng không rườm rà,…

**Tổng quan về thanh toán MOMO**

**Giới thiệu chung**

Giới thiệu chung Momo Payment Platform API là giải pháp thanh toán cho các đơn vị kinh doanh, cho phép khách hàng sử dụng tài khoản Ví MoMo để thanh toán các dịch vụ trên nhiều nền tảng khác nhau: Desktop Website, Mobile Website, Mobile Application, POS, Pay In Bill, Web In App MoMo.

Phương thức thanh toán gồm có 4 hình thức khác nhau:

* [**Cổng thanh toán MoMo (All In One)**](https://developers.momo.vn/#/docs/aio/)**:** Áp dụng đối tác có thanh toán trên nền tảng Website, Mobile, Smart TV,...
* [**Thanh toán App-In-App**](https://developers.momo.vn/#/docs/app_in_app)**:** Áp dụng cho đối tác có ứng dụng di động (android/ios) muốn mở trực tiếp ứng dụng MoMo để thanh toán.
* [**Thanh toán POS**](https://developers.momo.vn/#/docs/pos_payment)**:** Áp dụng cho đối tác có hệ thống bán hàng bằng máy POS. Thu ngân dùng máy scan để quét "**MÃ THANH TOÁN**" trên app MoMo để thanh toán
* [**Thanh toán QR Code**](https://developers.momo.vn/#/docs/qr_payment)**:** Đối tác tạo QR code theo định dạng MoMo cung cấp, khách hàng chỉ cần dùng app MoMo để quét và thanh toán.

Tất cả các API của **Cổng thanh toán MoMo** đều là riêng biệt. Không sử dụng chung với các API của **App-In-App**, **POS**, **QR Code**, ...

**Tài khoản doanh nghiệp**

Mỗi đối tác khi liên kết với MoMo sẽ sử dụng tài khoản doanh nghiệp. Tài khoản này sẽ dùng để quản lý các giao dịch, thông tin tích hợp, cửa hàng, sub-merchant, v.v..

**Thông tin tích hợp**

MoMo cung cấp cho đơn vị kinh doanh hai môi trường để tích hợp với MoMo API:

* **Testing**: Sử dụng trong quá trình tích hợp: xây dựng tính năng, kiểm thử, debug, v.v..
* **Production**: Sử dụng để triển khai cho [người](https://en.wikipedia.org/wiki/End_user) dùng cuối thanh toán dịch vụ.

Thông tin cấu hình để kết nối với**MoMo API**.

* **Partner Code**: Thông tin để định danh tài khoản doanh nghiệp.
* **Access Key**: Cấp quyền truy cập vào hệ thống MoMo.
* **Secret Key**: Dùng để tạo chữ ký điện tử signature.
* **Public Key**: Sử dụng để tạo mã hoá dữ liệu bằng thuật toán RSA.

Các thông tin này sẽ thay đổi theo từng môi trường.

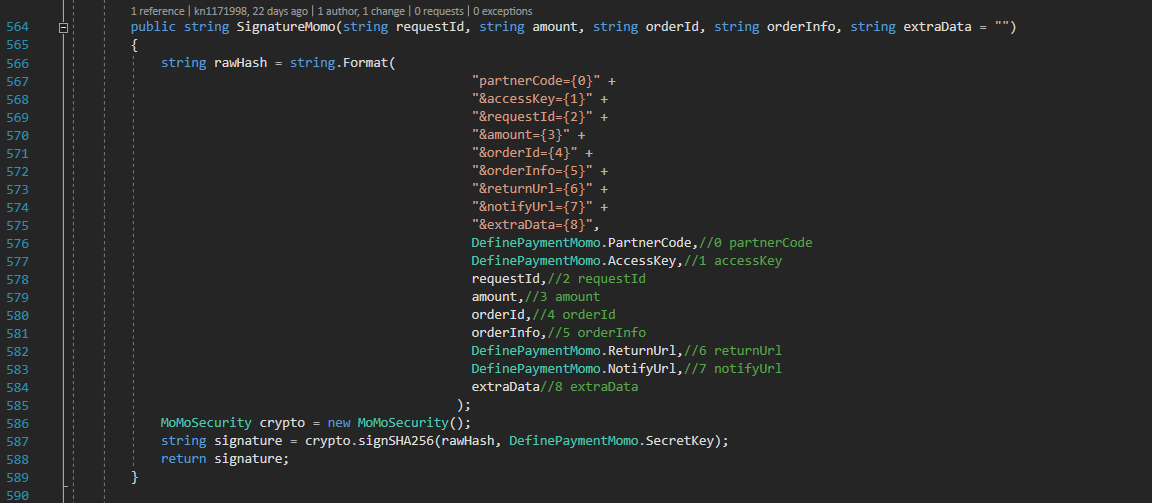
**Bảo mật**

**MoMo** sử dụng [chữ ký điện tử](https://en.wikipedia.org/wiki/Digital_signature) và [mã hoá dữ liệu](https://en.wikipedia.org/wiki/Encryption) để xác thực dữ liệu đầu vào và ra trên mỗi yêu cầu **HTTP Request**/**HTTP Response**.

Chữ ký điện tử:

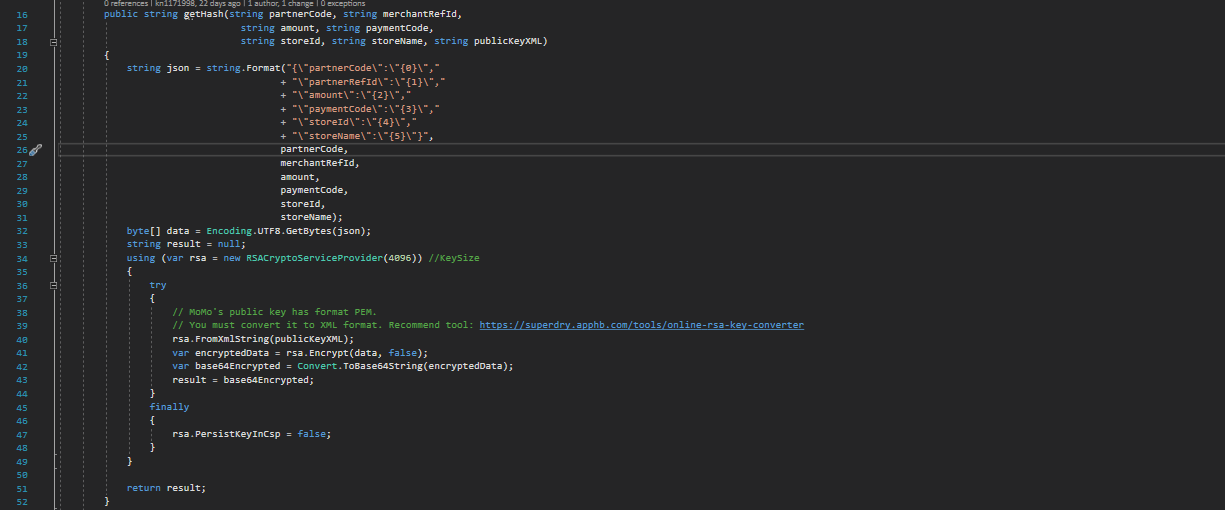
signature là một chuỗi ký tự được tạo ra từ một thuật toán cho trước, sử dụng để **kiểm tra tính đúng đắn của dữ liệu** trên đường truyền giữa 2 hệ thống. Một số thuật toán đang sử dụng là MD5, SHA1, SHA256 và Hmac.

Trong Momo sử dụng thuật toán **HMAC\_SHA256** để tạo signature. Dữ liệu đầu vào bao gồm [Secret Key](https://developers.momo.vn/#/home?id=key-credential) và data, data được tạo ra theo định dạng: key1=value1&key2=value2...

****Ví dụ viết bằng C#:

**Mã hoá RSA**

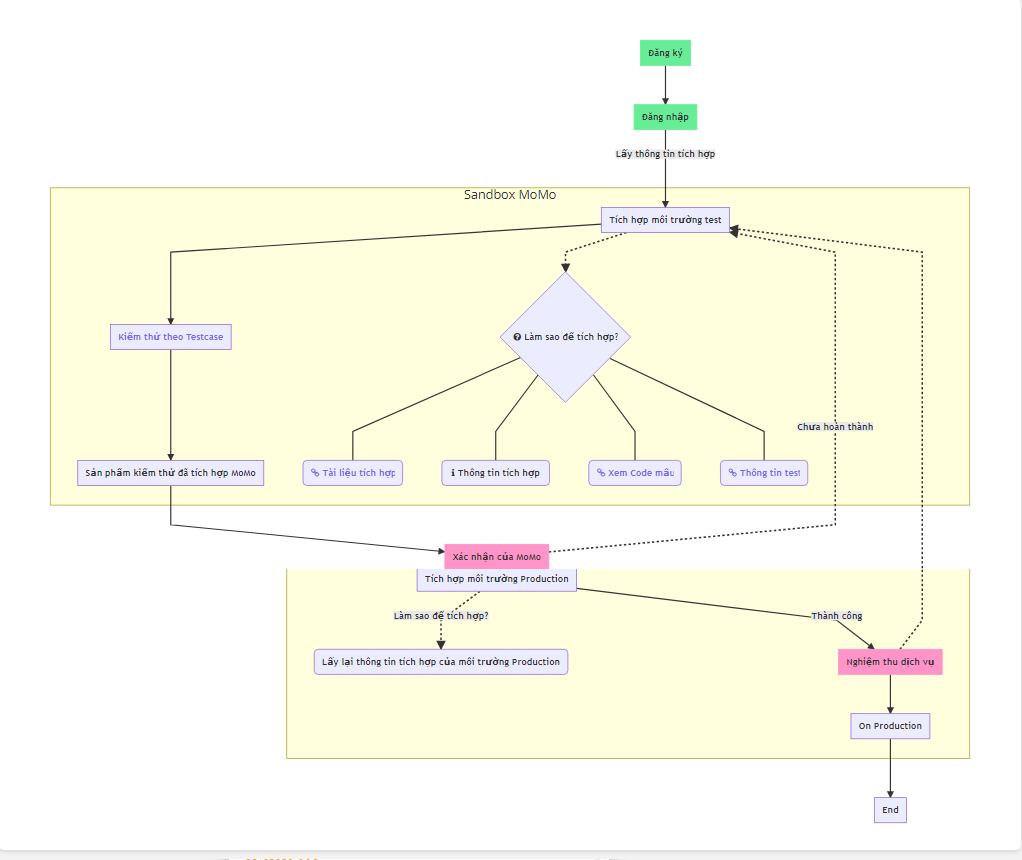
[Mã hóa RSA](https://vi.wikipedia.org/wiki/RSA_(m%C3%A3_h%C3%B3a)) là một thuật toán **mã hóa khóa công khai** để **bảo vệ thông tin** trên đường truyền. Sử dụng một cặp key (public key và private key) để mã hóa và giải mã dữ liệu. Đối tác dùng [public key](https://developers.momo.vn/#/home?id=key-credential) do MoMo cung cấp để mã hóa data theo định dạng của MoMo, MoMo sẽ giải mã bằng **private key**.

Ví dụ viết bằng C#:

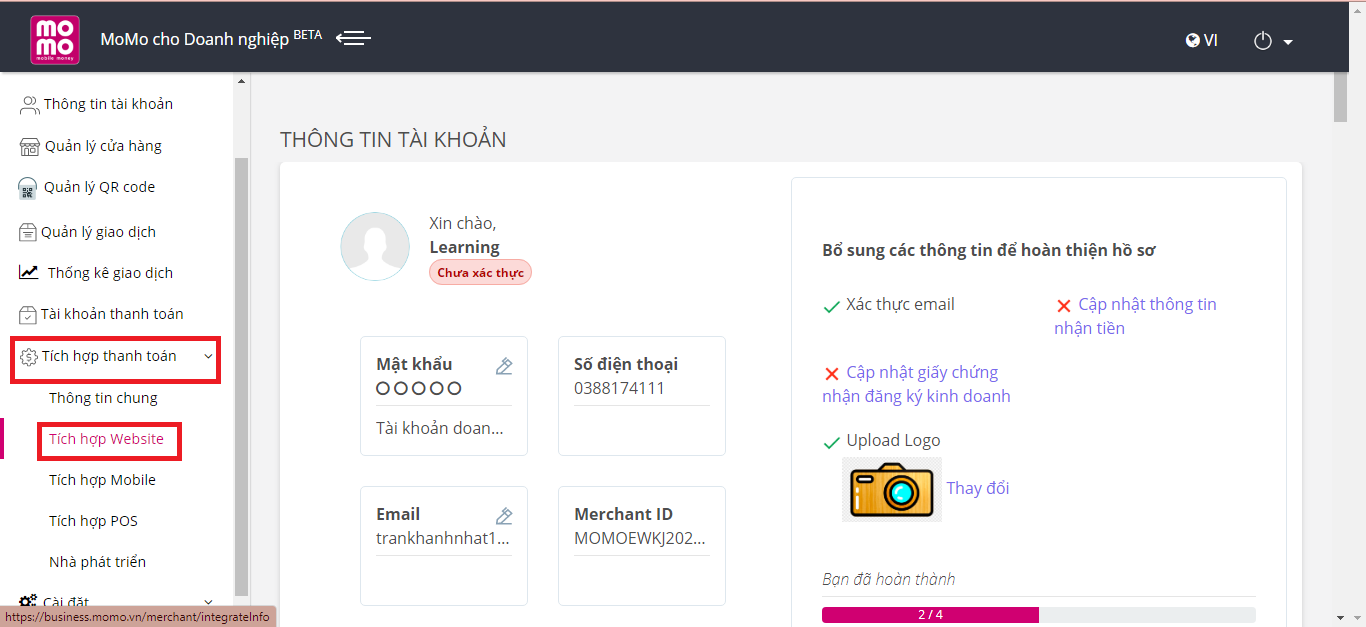
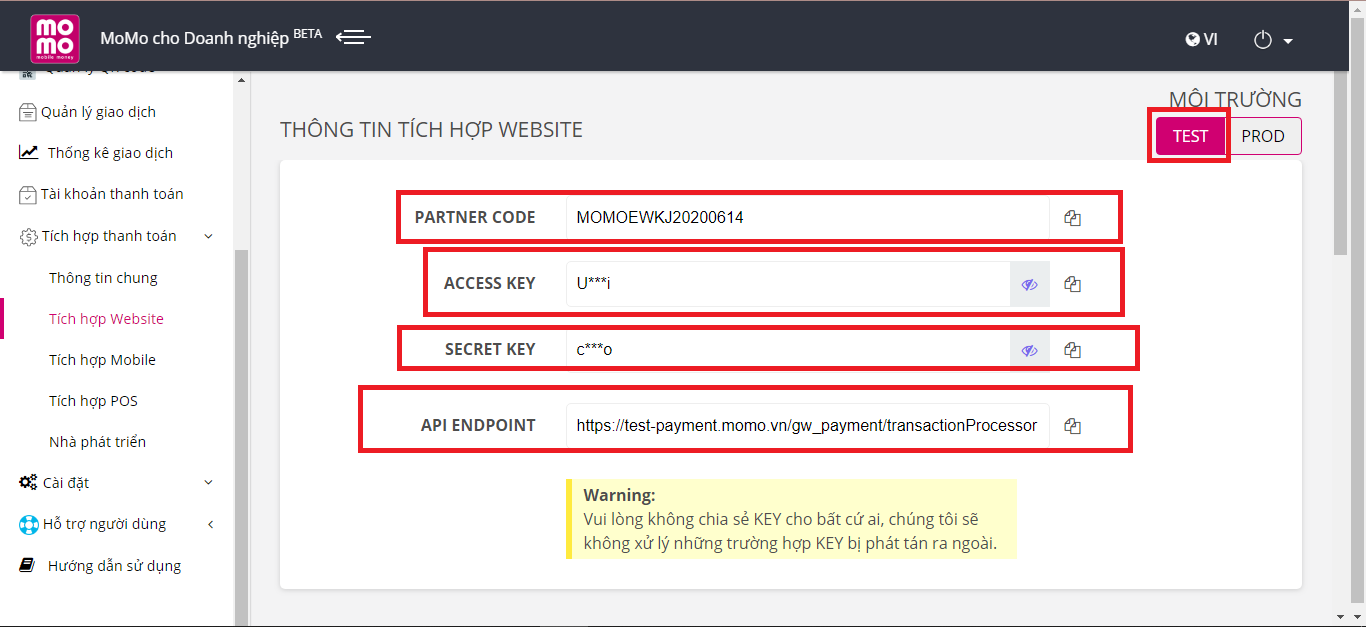
**Quy trình tích hợp**

* Do môi trường tích hợp của chúng ta là môi trường website nên sẽ sử dụng **Cổng thanh toán Momo(All In One)** để tích hợp.

Hình ảnh quy trình tích hợp



Các bước cơ bản để tích hợp với Momo:

* Đăng ký tài khoản doanh nghiệp trên website <https://business.momo.vn/> của Momo
* Sau khi đăng ký thì đăng nhập vào tài khoản doanh nghiệp
* Khi vào ta chọn vào mục tích hợp website như hình bên dưới:
* Sau đó ta chọn môi trường test để sử dụng trong quá trình lập trình, môi trường product được sử dụng khi dự án đã hoàn thành và đưa vào sử dụng thực tế.
* Ta sử dụng PartnerCode, Access Key, SecretKey và Api EndPoint để tích hợp vào website trong quá trình phát triển hệ thống của mình.

**Cổng thanh toán MOMO**

**Giới thiệu**

Momo All In One Payment là giải pháp thanh toán của Momo áp dụng trên nhiều nền tảng khác nhau chỉ trong một API duy nhất.

Sử dụng tài khoản thử nghiệm của Momo để trải nghiệm thanh toán.

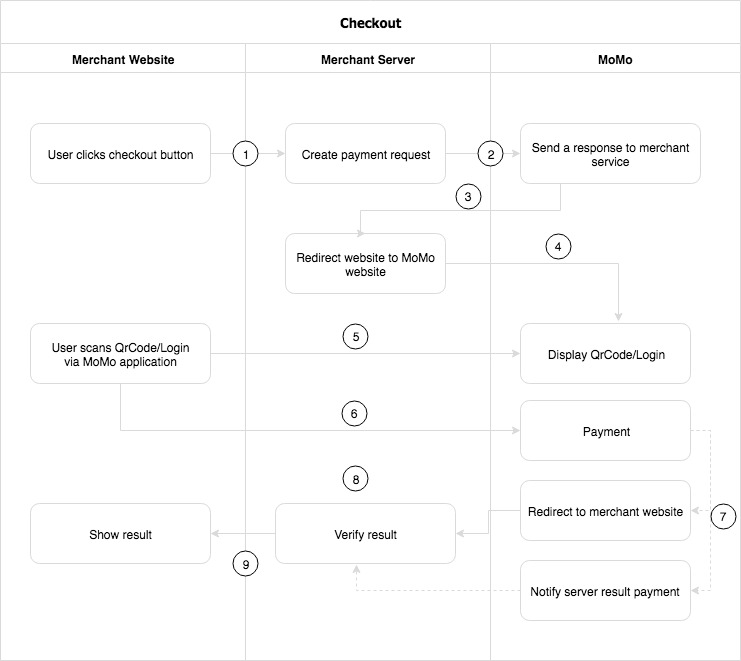
Sử dụng ứng dụng **MoMo TEST** và **tài khoản đăng nhập MoMo TEST** được cung cấp để thực hiện giao dịch.

Tài khoản đăng nhập

|  |  |  |  |
| --- | --- | --- | --- |
| Thành công | Hạn mức 20.000.000 | Hạn mức 5.000.000 | Không thành công |
| 0917003003 | 0919100100 | 0918002000 | 0916005000 |
| 0917030000 | 0919100010 | 0918002020 | 0916005050 |
| 0917030003 | 0919100001 | 0918002200 | 0916005500 |
| 0917030030 | 0919010100 |  |  |
| 0917030300 | 0919010010 |  |  |
| 0917300300 | 0919100101 |  |  |

* Mật khẩu: 000000
* Mã xác thực (OTP): 000000

**Sơ đồ xử lý**

Thanh toán đơn hàng trên website desktop/mobile

Mô hình quy trình thanh toán

* Bước 1: Khách hàng kiểm tra đơn hàng và chọn Momo là phương thức thanh toán.
* Bước 2: Server của bạn tạo session thanh toán và gửi yêu cầu thanh toán qua Momo.
* Bước 3: Chuyển trang mua hàng sang trang thanh toán của MoMo.
* Bước 4: Khách hàng sử dụng ứng dụng MoMo để quét mã QR hoặc đăng nhập để thanh toán
* Bước 5: Sau khi thanh toán MoMo sẽ chuyển khách hàng về trang mua hàng
* Bước 6: Server của bạn xác thực giao dịch và cập nhật dịch vụ cho khách hàng

<https://github.com/janl/mustache.js>

<https://techblog.vn/tim-hieu-ve-x-template-trong-jquery>

<https://completejavascript.com/mustache-template-voi-jquery>

[https://business.momo.vn/assets//docs/guide/merchant/User\_guide\_M4B\_merchant(VI).pdf](https://business.momo.vn/assets/docs/guide/merchant/User_guide_M4B_merchant(VI).pdf)

<https://developers.momo.vn/#/docs/testing_information>