TÀI LIỆU THIẾT KẾ PHẦN MỀM

HỆ THỐNG SIÊU THỊ QUẢN LÝ MUA BÁN THIẾT BỊ THÔNG MINH ESHOP

[KIẾN THỨC BỔ TRỢ(.Net Core) 3](#_Toc37615026)

[I. Hướng dẫn viết api và gen ra service cho angular 3](#_Toc37615027)

[PHẦN 1. TỔNG QUAN 3](#_Toc37615028)

[I. Phát biểu bài toán 3](#_Toc37615029)

[II. Các tính năng chính 3](#_Toc37615030)

[1. BÁN HÀNG ĐA KÊNH 3](#_Toc37615031)

[a. Bán hàng tại POS 3](#_Toc37615032)

[b. Kết nối các sàn thương mại điện tử như Tiki,Shopee 3](#_Toc37615033)

[c. Kênh google Shopping 3](#_Toc37615034)

[2. QUẢN TRỊ TẬP TRUNG OMNICHANNEL 3](#_Toc37615035)

[a. Quản lý sản phẩm 3](#_Toc37615036)

[b. Quản lý khách hàng 4](#_Toc37615037)

[c. Quản lý đơn hàng 4](#_Toc37615038)

[d. Quản lý khuyến mãi 4](#_Toc37615039)

[e. Quản lý tồn kho 4](#_Toc37615040)

[f. Quản lý thanh toán 4](#_Toc37615041)

[g. Quản lý vận chuyển 4](#_Toc37615042)

[h. Quản lý dữ liệu đơn hàng theo điểm bán 4](#_Toc37615043)

[i. Phân quyền quản trị cho nhân viên 4](#_Toc37615044)

[3. MÔ HÌNH VẬN HÀNH GIAO HÀNG 4](#_Toc37615045)

[a. Fulfillment Network 4](#_Toc37615046)

[4. CÁC GIẢI PHÁP MAKETING 4](#_Toc37615047)

[a. Maketing thông qua email 4](#_Toc37615048)

[b. Kết nối Facebook Pixel 4](#_Toc37615049)

[c. Tích hợp Facebook Messenger Livechat 4](#_Toc37615050)

[5. BÁO CÁO THỐNG KÊ HIỆU QUẢ KINH DOANH 4](#_Toc37615051)

[d. Báo cáo doang thu 4](#_Toc37615052)

[e. Báo cáo tồn kho 4](#_Toc37615053)

[f. Thống kê lượng truy cập 4](#_Toc37615054)

[g. Google analytic 4](#_Toc37615055)

[PHẦN 2. CƠ SỞ LÝ THUYẾT 4](#_Toc37615056)

[I. Công nghệ website 5](#_Toc37615057)

[1. Server .net core 3.1 5](#_Toc37615058)

[a. Giới thiệu về .NET core 5](#_Toc37615059)

[a. Các đặt trưng của .net core 5](#_Toc37615060)

[2. Client angular 8 5](#_Toc37615061)

[a. Giới thiệu về angular 5](#_Toc37615062)

[II. Mô hình website 6](#_Toc37615063)

[1. **RESTful Web Service** 6](#_Toc37615064)

[a. Web service là gì 6](#_Toc37615065)

[b. RESTful web service 6](#_Toc37615066)

[I. Google analytic 9](#_Toc37615067)

[2. Giới thiệu 9](#_Toc37615068)

[1. Tính năng 9](#_Toc37615069)

[**a.** **Thống kê thời gian thực (real-time)** 9](#_Toc37615070)

[**b.** **Thống kê nguồn truy cập của người dùng, ngôn ngữ họ sử dụng cùng hệ điều hành của thiết bị người dùng** 9](#_Toc37615071)

[**c.** **Chỉ rõ hành vi người dùng trên website** 9](#_Toc37615072)

[**d.** **Phân tích lưu lượng truy cập theo nhân khẩu học** 9](#_Toc37615073)

[PHẦN 3. PHÂN TÍCH ĐỀ TÀI 9](#_Toc37615074)

[PHẦN 4. THIẾT KẾ CÀI ĐẶT 9](#_Toc37615075)

[PHẦN 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 9](#_Toc37615076)

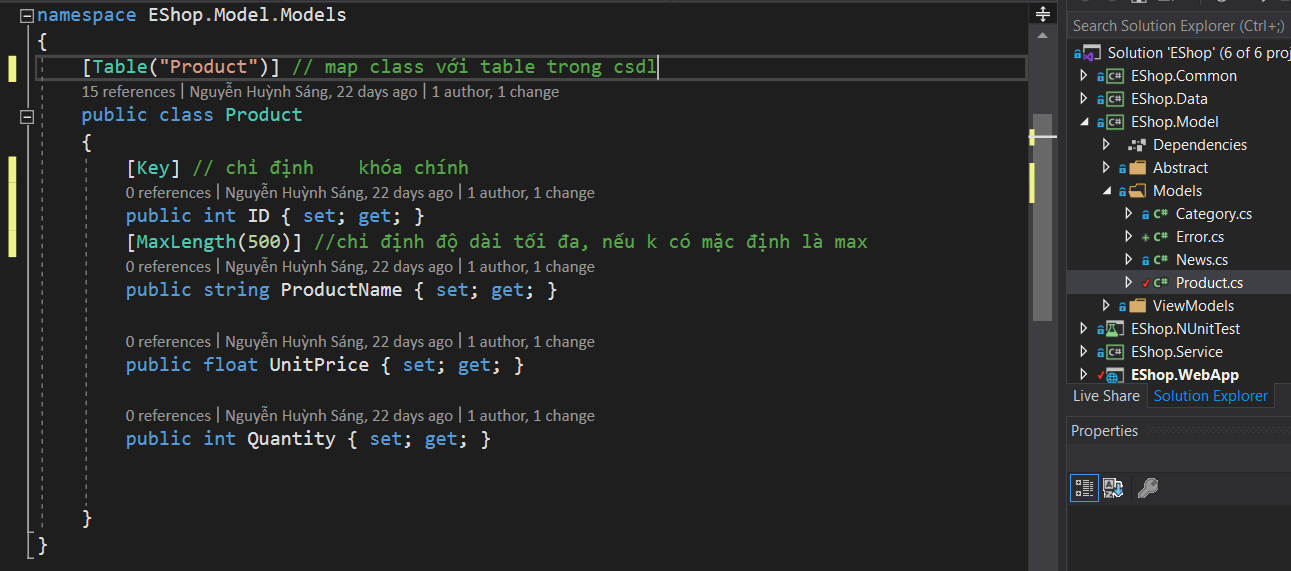
# KIẾN THỨC BỔ TRỢ(.Net Core)

## Hướng dẫn viết api và gen ra service cho angular

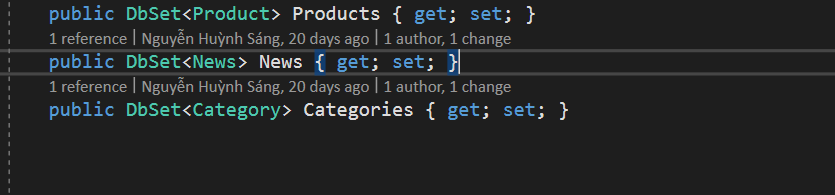
### Tạo bảng mới

Dựa vào csdl đã thiết kế

1. Mở project Eshop.Model/Models tạo 1 class mới trùng tên bảng định tạo



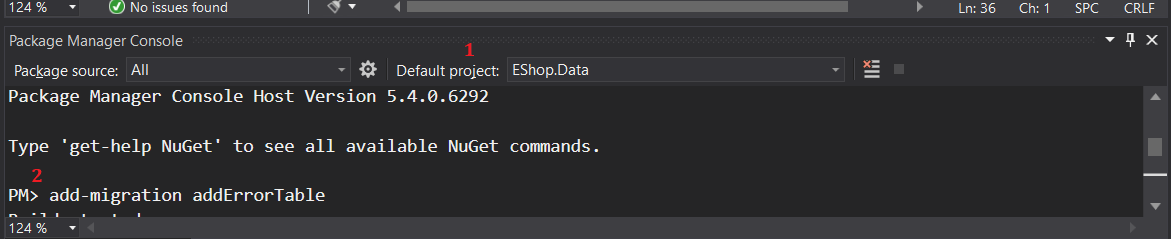
1. Mở Eshop.Data/EshopDbContext.cs tạo dataset mới với tên số nhiều



1. Mở Package Manager Console (tool/nuget package manager/ Package Manager Console) chọn (1) default project là eshop.data

Thêm dòng lệnh

add-migration {tên}

{tên}: nên đặt rõ ràng phần thêm gì hay sửa gì

Sau đó nhấn enter để tạo migration

1. Gõ

 Update-database

để cập nhật csdl vào sql

### Tạo reponsitory cho entity

1. Mở Eshop.Data/Repository tạo class repository cho entity mới

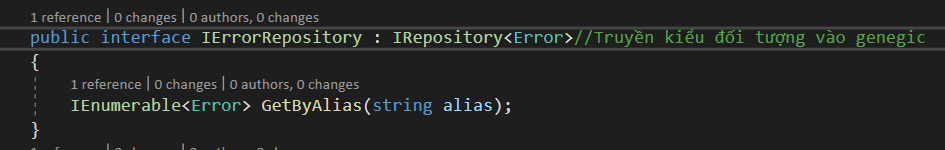
Vd: ErrorRepository

1. Tạo interface tương ứng với repository vừa tạo ngay trong class ở bước 1

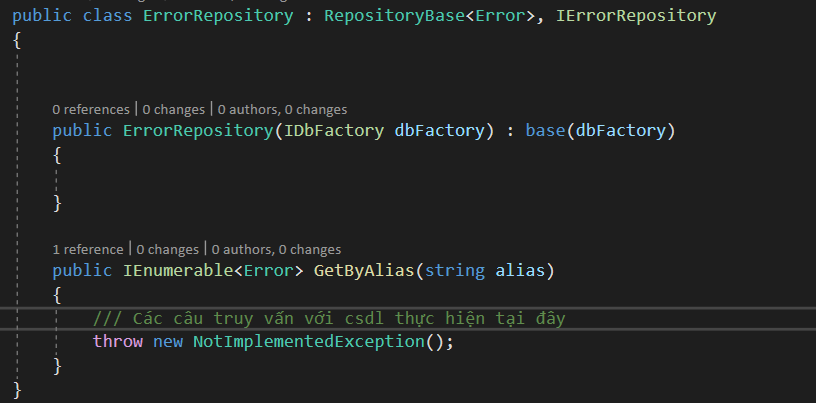
Vd: IErrorRepository

Interface này kế thừa từ interface Irepository

IErrorRepository: chứa các tên phương thức riêng của đối tượng và các phương thức như create ,update, delete… đã viết sẵn trong Irepository



1. Class Repository vừa tạo ở bước 1 kế thừa interface tạo ra ở bước 2 và class RepositoryBase,và ta định nghĩa các phương thức riêng implement từ interface tại đây



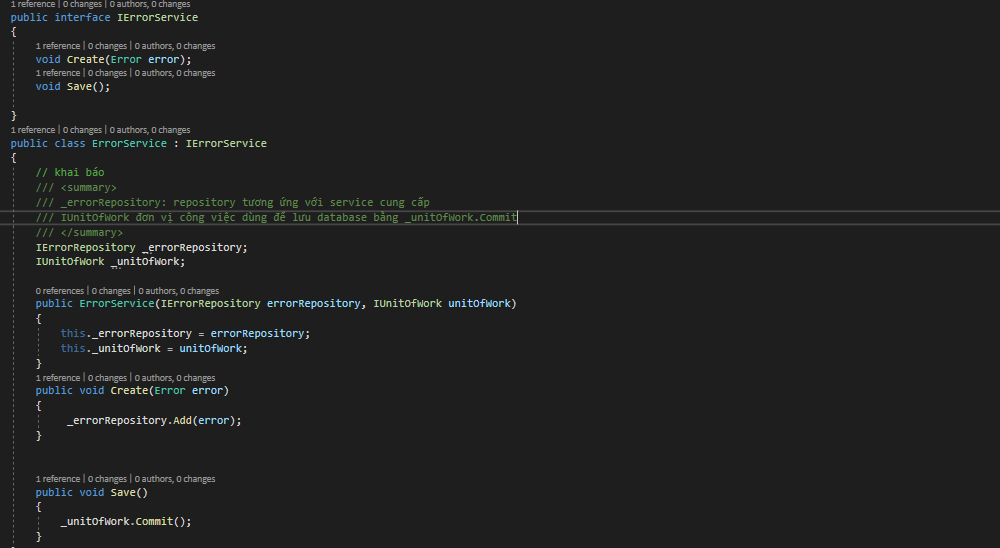
### Tạo service cho entity

Ta tương tác giữa controller và database thông qua service.

1. Vào Eshop.Service/Service tạo 1 service mới

Vd: ErrorService

1. Tạo Interface cho service vừa tạo ngay trong lớp đó, Interface này chứa tất cả các phương thức service cung cấp
2. class service vừa tạo ở bước 1 kế thừa lại interface của bước 2 và định nghĩa các phương thức, service sẽ dùng repository tưng ứng để giao tiếp với database



# TỔNG QUAN

Các nhóm sau thời gian tìm hiểu tài liệu những vấn đề liên quan đến đề tài, cái gì, ở đâu .v.v đề cập đến đề tài đang làm ==> đánh giá và có thể khái quát chung vấn đề mình định làm là gì

## Phát biểu bài toán

EShop là một cửa hàng buôn bán thiết bị thông minh với quy mô trung bình . Cửa hàng là nơi mua bán các loại mặt hàng về các thiết bị thông minh như điện thoại, laptop. Với cách quản lý trên giấy tờ hiện nay, công ty gặp khó khăntrong việc quản lý số lượng mặt hàng và trong công việc bán hàng như quản lý đơn hàng, quản lý doanh thu, quản lý nhân viên. Do đó, cửa hàng cần một chương trình máy tính hiệu quảđể giải quyết tình trạng như hiện nay. Chương trình này cho phép kiểm soát tình hình mua bán và số lượng các mặt hàng xuất nhập kho theo định kỳ. Cửa hàng yêu cầu xây dựng một hệ thống cho phép người quản lý và nhân viên sử dụng phần mềm với các quyền đăng nhập khác nhau để giải quýêt các công việc khác nhau hiệu quả. Ngoài ra, khách hàng cũng có thể sử dụng hệ thống nhưng chỉ với chức năng tìm kiếm mặt hàng và có thể đặt hàng trên mạng từ xa.

## Các tính năng chính

Để có thể cạnh tranh với các đối thủ khác trên thị trường Eshop áp dụng các chiến lược sau

### BÁN HÀNG ĐA KÊNH

#### Bán hàng tại POS

#### Kết nối các sàn thương mại điện tử như Tiki,Shopee

#### Kênh google Shopping

### QUẢN TRỊ TẬP TRUNG OMNICHANNEL

Bán hàng đa kênh – Omni-channel không đơn thuần là việc bán hàng ở nhiều nơi khác nhau. Với mô hình Omni-channel, sản phẩm kinh doanh được đồng bộ trên các kênh bán hàng và hoạt động trơn tru trên 1 hệ thống quản lý. Thúc đẩy khách hàng mua sắm nhiều hơn, trải nghiệm liền mạch ở bất cứ đâu dù khách hàng online bằng di động hay laptop, trên sàn thương mại điện tử hay tại cửa hàng.

#### Quản lý sản phẩm

Sản phẩm được chia thành danh mục đa cấp với đầy đủ các đặt tính riêng có thể bán theo giá riêng tương ứng với từng đặt tính

#### Quản lý khách hàng

#### Quản lý đơn hàng

#### Quản lý khuyến mãi

#### Quản lý tồn kho

#### Quản lý thanh toán

Tích hợp các cổng thanh toán phổ biến như

* Ngân lượng
* Ví điện tử momo

#### Quản lý vận chuyển

#### Quản lý dữ liệu đơn hàng theo điểm bán

#### Phân quyền quản trị cho nhân viên

### MÔ HÌNH VẬN HÀNH GIAO HÀNG

#### Fulfillment Network

Dịch vụ fulfillment là quá trình bắt đầu từ lúc hàng hóa được nhập vào kho đến lúc người mua hàng nhận được sản phẩm, bao gồm các hoạt động lấy hàng từ người bán hàng, lưu kho, xử lý đơn hàng, lấy hàng từ kho, đóng gói và vận chuyển đến đúng địa chỉ của khách hàng. Nói cách khác, fulfillment thay người bán hàng làm tất cả công việc liên quan đến quản lý tồn kho, xử lý đơn hàng và vận chuyển, đảm bảo sản phẩm đến tay khách hàng (người mua hàng) nhanh chóng hơn. Fulfillment còn được gọi bằng nhiều cái tên khác nhau như dịch vụ hoàn tất đơn hàng, trung tâm phân phối sản phẩm hoặc dịch vụ hậu cần kho vận.

### CÁC GIẢI PHÁP MAKETING

#### Maketing thông qua email

#### Kết nối Facebook Pixel

#### Tích hợp Facebook Messenger Livechat

### BÁO CÁO THỐNG KÊ HIỆU QUẢ KINH DOANH

#### Báo cáo doang thu

#### Báo cáo tồn kho

#### Thống kê lượng truy cập

#### Google analytic

# CƠ SỞ LÝ THUYẾT

Sau khi đã cân nhắc cái mình định làm có liên quan đến lý thuyết nào, thuật toán nào, phương pháp nào, công nghệ gì v.v thì có thể trình bày ở đây

## Công nghệ website

### Server .net core 3.1

#### Giới thiệu về .NET core

.NET Core là môi trường thực thi. Nó được thiết kế lại hoàn toàn của .NET Framework. Mục tiêu chính của .NET Core là hỗ trợ phát triển ứng dụng đa nền tảng cho ứng dụng .NET. Nó được hỗ trợ trên Windows, Mac OS và Linux. .NET Core là một framework mã nguồn mở được xây dựng và phát triển bởi Microsoft và cộng đồng .NET trên Github

NET Core là một tập con của Full .NET Framwork. WebForms, Windows Forms, WPF không phải là một phần của .NET Core.

Nó cũng triển khai đặc điểm của .NET Standard.

#### Các đặt trưng của .net core

* Bạn có thể xây dựng và chạy ứng dụng ASP.NET đa nền tảng trên Windows, Mac và Linux (mã nguồn mở và cộng đồng phát triển)
* ASP.NET Core hợp nhất ASP.NET MVC và ASP.NET Web API.
* Có thể host trên IIS hoặc tự host.
* Có sẵn Dependency Injection.
* Dễ dàng tích hợp với các framework frontend như Angular, Knockout...
* Hỗ trợ cấu hình cho nhiều môi trường.
* Cơ chết HTTP Request pipeline mới.
* Hỗ trợ quản lý phiên bản
* Dùng chung toàn bộ Nuget Package.

### Client angular 8

#### Giới thiệu về angular

Angular 2+ là một javascript framework để xây dựng các application web bằng JavaScript , HTML và TypeScript , là một supperset của JavaScript . Angular cung cấp các tính năng tích hợp cho animation , http service và các tài liệu lần lượt có các tính năng như auto-complete , navigation , toolbar , menus ,... Code được viết bằng TypeScript , biên dịch thành JavaScript và hiển thị hầu như viết lại hoàn toàn của Angularjs . Controllers và $scope ( Angular 1 ) được thay thể bởi components và directives . Components = directives + template , tạo nên view của ứng dụng và xử lí các logic trên view . Thay vì viết một file HTML dài và định nghĩa các event trong các html component như trong Angular v1 thì view được chia thành nhiều component với khai báo class từ đó giúp dev dễ dàng tái sử dụng mã nguồn. hị tương tự trong trình duyệt.

## Mô hình website

### **RESTful Web Service**

#### Web service là gì

* **Web service** là một dịch vụ mà nó kết hợp các máy tính cá nhân với các thiết bị khác, các cơ sở dữ liệu và các mạng máy tính để tạo thành một cơ cấu tính toán ảo mà người sử dụng có thể làm việc thông qua các trình duyệt mạng. Bản thân các dịch vụ này sẽ chạy trên các máy chủ trên nền Internet. Các máy chủ của một nhà cung cấp dịch vụ web này cũng cần trở thành nguồn cung cấp cho người sử dụng cả về độ an toàn, độ riêng tư và khả nǎng truy nhập.
* Các **Web Service** thường trả về dữ liệu dưới dạng máy tính có thể đọc được là **XML** hoặc **JSON** rồi trả về trình duyệt ở phía client.
* Các công nghệ thường được sử dụng để xây dựng web service:
  + **SOAP (Simple Object Access Protocol)**
  + **WSDL (Web Services Description Language)**
  + **REST**

#### RESTful web service

RESTful web service là các dịch vụ web được xây dựng dựa trên cấu trúc REST (REpresentational State Transfer). Tức là nó giống như một kiến trúc, nguyên tắc cần tuân theo để thiết kế, xây dựng một web service.

Trong kiến trúc REST mọi thứ đều được coi là tài nguyên, chúng có thể là: tệp văn bản, ảnh, trang html, video, hoặc dữ liệu động… REST server cung cấp quyền truy cập vào các tài nguyên, REST client truy cập và thay đổi các tài nguyên đó. Ở đây các tài nguyên được định danh dựa vào URI, REST sử dụng một vài đại diện để biểu diễn các tài nguyên như văn bản, JSON, XML.

Và nội dung của kiến trúc REST bao gồm bốn nguyên tắc cơ bản sau:

**Cách sử dụng phương thức HTTP**

* Như chúng ta đã biết, HTTP cung cấp các phương thức dùng để lấy dữ liệu, trèn dữ liệu, cập nhập dữ liệu hoặc xóa dữ liệu. Và khi sử dụng những phương thức này, chúng ta cần xác định rõ ràng mục đích sử dụng mỗi khi gọi tới một phương thức. Và gợi ý cụ thể cho các phương thức như sau:
  + **GET**: dùng để truy xuất một tài nguyên (phương thức này gần như là phổ biến nhất)
  + **POST**: dùng để tạo một tài nguyên trên máy chủ (VD như đăng kí tài khoản, sau khi điền form thông tin, dùng phương thức POST để gửi dữ liệu lên máy chủ)
  + **PUT**: dùng để thay đổi trạng thái một tài nguyên hoặc để cập nhật nó.
  + **DELETE**: dùng để huỷ bỏ hoặc xoá một tài nguyên.

**Phi trạng thái (stateless)**

* Phi trạng thái có nghĩa là máy chủ sẽ không lưu giữ thông tin của client mà nó giao tiếp, thông tin hoặc được giữ trên client hoặc được chuyển thành trạng thái của tài nguyên. Mỗi request lên server thì client phải đóng gói thông tin đầy đủ để thằng server hiểu được.
* Điểu này đem lại hai lợi ích:
  + Giúp tách biệt client ra khỏi sự thay đổi của server.
  + Giúp hệ thống của bạn dễ phát triển,bảo trì, mở rộng vì không cần tốn công CRUD trạng thái của client.
* VD:

Bạn vừa gửi yêu cầu để xem trang thứ 2 của một tài liệu.

Bây giờ bạn muốn xem trang tiếp theo (sẽ là trang 3).

REST không lưu trữ lại thông tin rằng trước đó nó đã phục vụ bạn trang số 2. Điều đó có nghĩa là REST không quản lý phiên làm việc (Session)  
--> Máy chủ chuyển hầu hết vai trò duy trì trạng thái sang ứng dụng ở máy khách, từ đó giúp cho nó hoạt động tốt hơn.

**Hiển thị cấu trúc thư mục như URI**

* REST đưa ra một cấu trúc để người dùng có thể truy cập vào tài nguyên của nó thông qua các URL
* Các địa chỉ REST service cần phải thật trực quan đến mức đơn giản, có thể dự đoán, và dễ hiểu. Ví dụ: chỉ cần nhìn vào thanh địa chỉ URL ta có thể đoán rằng nó đang trỏ tới cái gì và cung cấp tài nguyên gì.
* Và để tạo ra đáp ứng yêu cầu trên thì ta nên định nghĩa URI có câu trúc giống thư mục. Loại URI này có phân cấp, có gốc là một đường dẫn đơn, các nhánh từ gốc là các đường dẫn phụ dẫn đến các các vùng service chính.

--> cấu trúc này giúp cho nhà phát triển dễ dàng trong việc cài đặt service của mình hướng vào một loại tài nguyên cụ thể nào đó.

* Ngoài ra còn có một số quy tắc bổ sung trong khi nói về cấu trúc địa chỉ của **RESTful Web service**:
  + Giấu các đuôi tài liệu mở rộng của bản gốc trong máy chủ (.jsp, .php, .asp), nếu có, vì vậy bạn có thể giấu một số thứ mà không cần thay đổi địa chỉ Urls.
  + Để mọi thứ là chữ thường.
  + Thay thế các khoảng trống bằng gạch chân hoặc hoặc gạch nối (một trong hai loại).
  + Thay vì sử dụng mã (404 Not Found) khi yêu cầu địa chỉ cho một phần đường dẫn, luôn luôn cung cấp một trang mặc định hoặc tài nguyên như một phản hồi.

**Định dạng dữ liệu (html, json, text, xml…)**

* Khi Client gửi một yêu cầu tới web service, nó thường được truyền tải dưới dạng dữ liệu mà máy tính hiểu được (XML hoặc JSON) và thông thường nhận về với hình thức tương tự từ máy chủ.
* Tuy nhiên Client cũng có thể chỉ định kiểu dữ liệu nhận về mà nó mong muốn (JSON, hoặc XML,..), các chỉ định này được gọi là các kiểu MIME, nó được gửi kèm trên phần HEADER của request.
* Một số kiểu MIME phổ biến:

| **MIME-TYPE** | **Content-Type** |
| --- | --- |
| JSON | application/json |
| XML | application/xml |
| XHTML | application/xhtml+xml |

## Google analytic

### Giới thiệu

Google Analytics là tool trực tuyến cho các nhà quảng cáo có thể theo dõi số liệu liên quan đến hoạt động của website. Với mong muốn đem đến môi trường Internet “xanh – sạch – đẹp”, Google cung cấp ứng dụng Analytics này nhằm hỗ trợ các quản trị viên website có thể đánh giá tổng thể tình trạng của website với chỉ một con bot tuần tra. **Google Analytic** đảm bảo số liệu về website mà họ cung cấp đều chính xác.

### Tính năng

#### **Thống kê thời gian thực (real-time)**

Google Analytic giúp bạn thấy được có bao nhiêu người dùng đang lướt website của bạn. Qua đó giúp bạn nắm rõ lưu lượng truy cập tối ưu nhất của website trong 1 ngày, tạo tiền đề để xây dựng [KPIs](https://vi.wikipedia.org/wiki/Qu%E1%BA%A3n_l%C3%BD_ch%E1%BA%A5t_l%C6%B0%E1%BB%A3ng) cho chiến dịch Digital Marketing của bạn.

#### **Thống kê nguồn truy cập của người dùng, ngôn ngữ họ sử dụng cùng hệ điều hành của thiết bị người dùng**

Google Analytics thống kê được nguồn truy cập vào website của người dùng đến từ đâu là nhiều nhất, cụ thể bao gồm các kênh như social media, google search, quảng cáo, các website khác… Ngoài ra, nó còn có khả năng thống kê được ngôn ngữ và hệ điều hành mà thiết bị của người dùng đang sử dụng để truy cập vào website của bạn là gì, từ đó tạo tiền đề để bạn [tối ưu website](https://mona.media/seo-la-gi-va-tam-quan-trong-trong-thuong-mai-dien-tu/) phù hợp với các tiêu chí trên.

#### **Chỉ rõ hành vi người dùng trên website**

Không chỉ dừng lại ở nguồn truy cập, **Google Analytics** còn cho thấy hành vi thực sự của họ trên website thông qua các chỉ số như thời gian trung bình của 1 phiên truy cập, trang được truy cập nhiều nhất trên website, tỷ lệ thoát trang cùng nhiều chỉ số khác mà bạn mong muốn tìm hiểu cụ thể hơn.

#### **Phân tích lưu lượng truy cập theo nhân khẩu học**

Google Analytics có thể phân tích các số liệu theo giới tính, địa điểm, sở thích… Điều này được Google thu thập thông qua cookies người dùng hoặc được máy chủ Google định vị, theo dõi.

# PHÂN TÍCH ĐỀ TÀI

Trình bày khảo sát phân tích, nêu được yêu cầu bài toán mình định làm gồm gì .v.v

# THIẾT KẾ CÀI ĐẶT

Nêu thiết kế có như các mô hình UML

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Sau thời gian tìm hiểu em thu được kết quả gì ... hạn chế gì để từ đó đưa ra hướng phát triên