



Bài 4

Tổng quan ứng dụng web

Module: Web Backend Development with NodeJS

Mục tiêu



- Hiểu được tổng quan mô hình web.
- Hiểu được mô hình Client - Server.
- Hiểu được cách hoạt động của giao thức HTTP.
- Hiểu được các phương thức của HTTP Requests.
- Hiểu được cấu trúc của HTTP Response.
- Tìm hiểu khái niệm Web server và làm việc được với module http trong nodejs.

Thảo luận

Web là gì?

Tổng quan mô hình Web.

Web và HTML.



- Web là công nghệ trên Internet cho phép thể hiện các thông tin một cách sinh động, gần gũi với con người hơn trên 1 trang thông tin gọi là trang Web. Trang Web được trình bày trên các bộ duyệt Web (Browser) trên các máy Client. Trang Web chính là các file văn bản dạng Text được cấu trúc hoá theo ngôn ngữ HTML.
- HTML viết tắt của Hypertext Markup Language là ngôn ngữ lập trình dùng để xây dựng và cấu trúc lại các thành phần có trong Website.

Các loại trang web.



- Trang Web tĩnh (trang .HTML) : Là trang đã được lưu trữ trên Server từ trước. Khi có yêu cầu từ phía Client, trang này sẽ được tải về Client và được hiển thị trên Server .
- Trang Web động (DHTML : Dynamic HTML) : Là các file chương trình được chạy trên Server. Sau khi chạy xong, Server sẽ sinh ra trang Web tĩnh HTML tương ứng với chương trình đó và gửi về Client. Nh vậy đến đây ta có thể coi rằng trang Web động là các trang Web HTML không tồn tại từ trước trên Server mà trái lại tùy theo yêu cầu từ phía Client, trang Web này sẽ được tạo ra sau khi chạy chương trình trên Server. Người dùng có thể thay đổi nội dung trang web bằng các thao tác trên trình duyệt để gửi HTTP requests đến server.

HTTP, Client, Server



- Http (HyperText Transfer Protocol) là giao thức truyền tải siêu văn bản được sử dụng trong www dùng để truyền tải dữ liệu giữa Web server đến các trình duyệt Web và ngược lại.
- Máy tính đóng vai trò là máy khách – Client: Với vai trò là máy khách, chúng sẽ không cung cấp tài nguyên đến các máy tính khác mà chỉ sử dụng tài nguyên được cung cấp từ máy chủ. Một client trong mô hình này có thể là một server cho mô hình khác, tùy thuộc vào nhu cầu sử dụng của người dùng.
- Máy tính đóng vai trò là máy chủ – Server: Là máy tính có khả năng cung cấp tài nguyên và các dịch vụ đến các máy khách khác trong hệ thống mạng. Server đóng vai trò hỗ trợ cho các hoạt động trên máy khách client diễn ra hiệu quả hơn.
- Client và server sẽ giao tiếp với nhau thông qua giao thức HTTP (HTTP Requests và HTTP Response).

Mô hình Client - Server

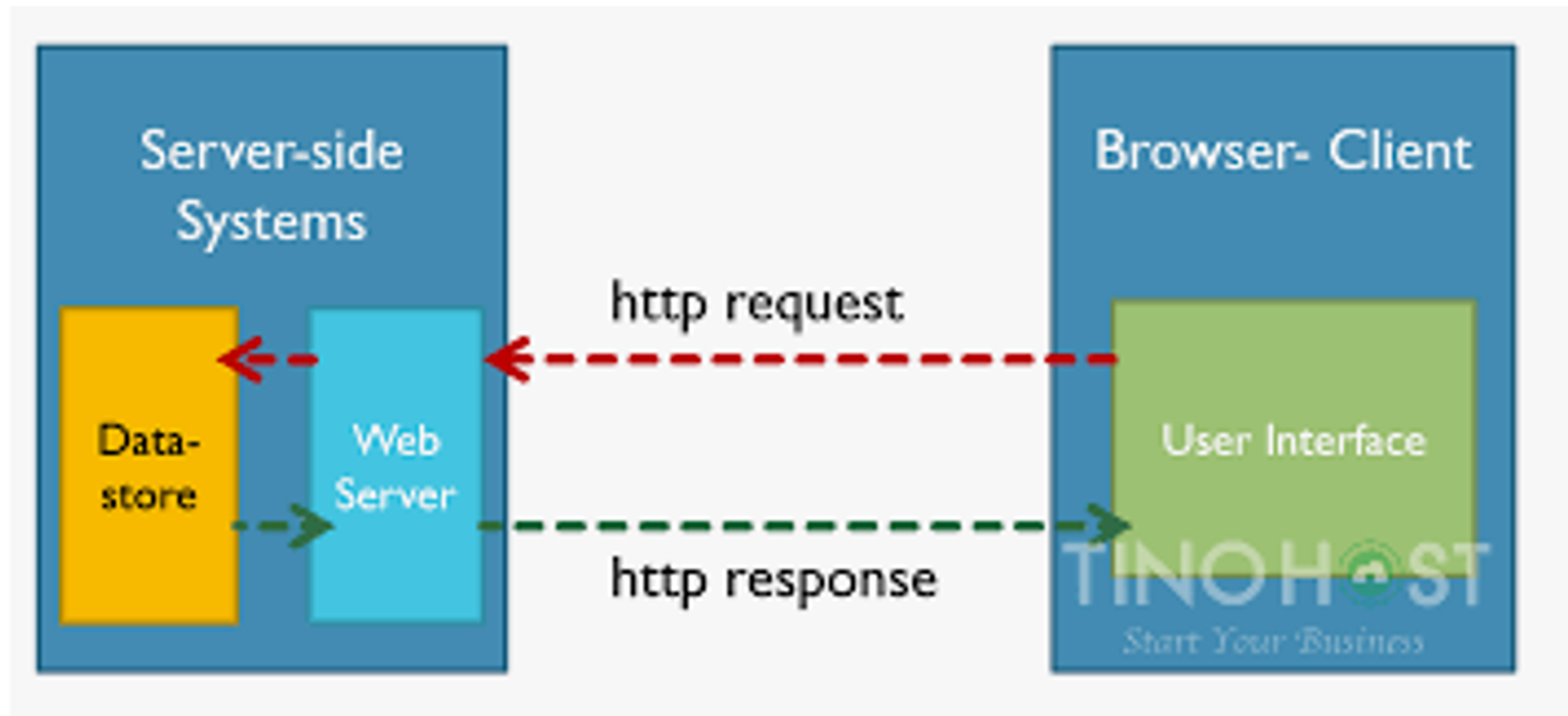
Các thành phần chính của mô hình Client - Server ?

Mô hình Client - Server



Một hệ thống web thông thường sẽ gồm 2 thành phần chính:

- Client (trình duyệt) là chương trình chạy trên thiết bị của người dùng (Chrome, cốc cốc ...).
- Web server là một máy chủ hay một phương tiện phục vụ các dịch vụ nào đó. Trong lĩnh vực công nghệ thì Server là một máy tính từ xa. Nó cung cấp các thông tin (dữ liệu) hoặc quyền truy cập vào các dịch vụ cụ thể.
- Client và web server tương tác với nhau qua một mạng truyền thông (mạng tcp/ip), sử dụng giao thức HTTP (hoặc HTTPS).



Cách Client - Server hoạt động



- Đầu tiên client sẽ gửi 1 http requests tới web server bằng cách thao tác trên trình duyệt (Truy cập tên miền trở tới server, login tài khoản ...).
- Server tiếp nhận requests và xử lý logic, kết nối đến cơ sở dữ liệu (Database) để lấy dữ liệu và trả về 1 http response cho client.
- Client nhận dữ liệu từ http response và hiển thị lên trình duyệt cho người dùng.

Ưu điểm của mô hình Client - Server



- Client server có khả năng chống quá tải mạng
- Client server đảm bảo toàn vẹn dữ liệu khi có sự cố xảy ra
- Dễ dàng mở rộng hệ thống mạng
- Chỉ cần chung định dạng giao tiếp mà không cần chung nền tảng là có thể hoạt động được
- Client server cho phép tích hợp các kỹ thuật hiện đại như GIS, mô hình thiết kế hướng đối tượng,...
- Với mô hình Client server, người dùng có thể truy cập dữ liệu từ xa, thực hiện các thao tác gửi, nhận file hay tìm kiếm thông tin đơn giản

Nhược điểm của mô hình Client - Server



- Cần bảo trì, bảo dưỡng server thường xuyên.
- Khả năng bảo mật thông tin mạng là một hạn chế nữa của Client server. Bởi vì, nguyên lý hoạt động của Client server là trao đổi dữ liệu giữa server và client ở 2 khu vực địa lý khác nhau. Trong quá trình trao đổi dữ liệu, khả năng thông tin mạng bị lộ là điều dễ xảy ra.



Tổng quan về HTTP

HTTP là gì?

HTTP là gì?

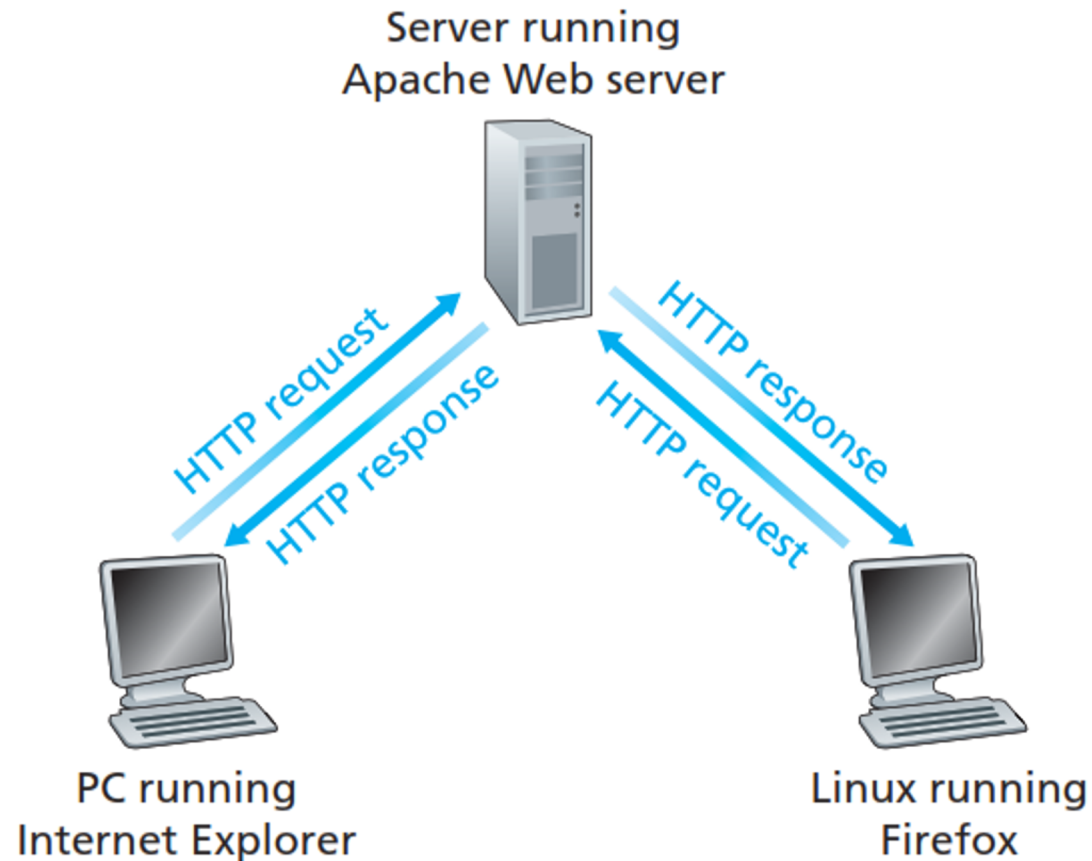


- HTTP (Hyper Text Transfer Protocol) là một giao thức nằm ở tầng ứng dụng (Application layer) của tập giao thức TCP/IP, sử dụng để truyền nhận dữ liệu giữa các hệ thống phân tán thông qua internet, cụ thể giao thức hoạt động theo mô hình Client-Server bằng cách thực hiện các quá trình request-response giữa các hệ thống máy tính khác nhau.
- Giao thức HTTP quy định cấu trúc của các gói tin và cách thức truyền nhận dữ liệu giữa client và server thông qua môi trường internet. Với khả năng truyền dẫn siêu văn bản (text, hình ảnh, âm thanh, video,...), HTTP hiện là nền tảng truyền dẫn dữ liệu của ứng dụng duyệt web hiện nay và được ứng dụng rất nhiều trong các hệ thống Internet of Things.

Cách HTTP hoạt động



Khi bạn gõ vào 1 địa chỉ vào trình duyệt Web, lúc này trình duyệt Web sẽ gửi 1 yêu cầu qua giao thức Http đến Web server. Web server sẽ nhận yêu cầu này và trả lại kết quả cho trình duyệt Web.



Quá trình của một phiên làm việc HTTP diễn ra như sau:

- HTTP client thiết lập một kết nối TCP đến server. Nếu thiết lập thành công, client và server sẽ truyền nhận dữ liệu với nhau thông qua kết nối này, kết nối được thiết lập còn gọi là socket interface bao gồm các thông tin: địa chỉ IP, loại giao thức giao vận (chính là TCP), và port (mặc định là 80).
- Sau khi kết nối thành công, client gửi một HTTP request đến server thông qua socket interface vừa được thiết lập. Trong gói tin request sẽ chứa đường dẫn yêu cầu (path name) là /category/internet-of-thing.
- Server sẽ nhận và xử lý request từ client thông qua socket, sau đó đóng gói dữ liệu tương ứng và gửi một HTTP response về cho client. Dữ liệu trả về sẽ là một file HTML chứa các loại dữ liệu khác nhau như văn bản, hình ảnh,...
- Server đóng kết nối TCP.
- Client nhận được dữ liệu phản hồi từ server và đóng kết nối TCP.

HTTPS là gì?



- Https (HyperText Transfer Protocol Secure) là giao thức Http có sử dụng thêm SSL (Secure Sockets Layer) để mã hóa dữ liệu trong lúc truyền tải dữ liệu nhằm gia tăng thêm tính an toàn cho việc truyền dữ liệu giữa Web server và trình duyệt Web.
- Giao thức Https thì sử dụng cổng 433 để truyền dữ liệu.



HTTP Requests

HTTP Requests là gì?

HTTP Requests là gì?



Trong mô hình client - server thì client sẽ gửi 1 http requests tới web server và server sẽ trả về http response.

HTTP request là thông tin được gửi từ client lên server, để yêu cầu server tìm hoặc xử lý một số thông tin, dữ liệu mà client muốn. HTTP request có thể là một file text dưới dạng XML hoặc Json mà cả hai đều có thể hiểu được.

Các phương thức của HTTP Requests



1. Phương thức GET

- Câu truy vấn sẽ được đính kèm vào đường dẫn HTTP request.
Ví dụ: `/?username="tinohost"&pass="tenmien"`
- GET request có thể được cached, bookmark và lưu trong lịch sử của trình duyệt mà bị giới hạn về chiều dài (chiều dài của URL là có hạn).
- Lưu ý: Bạn không nên dùng GET request với dữ liệu quan trọng mà chỉ dùng để nhận dữ liệu, không có tính bảo mật.

2. Phương thức POST.

- Câu truy vấn sẽ được gửi trong phần message body của HTTP request.
- POST không thể, cached, bookmark hay lưu trong lịch sử trình duyệt và cũng không bị giới hạn về độ dài.

3. Các phương thức khác.

- HEAD: tương tự như GET nhưng chỉ gửi về HTTP header.
- PUT: tải lên một mô tả về URL định trước.
- DELETE: xóa một tài nguyên định trước.
- OPTIONS: trả về phương thức HTTP mà server hỗ trợ.
- CONNECT: chuyển kết nối của HTTP request thành một kết nối HTTP tunnel.

Cấu trúc của HTTP Requests.



- Request line: Request line là dòng đầu tiên trong HTTP request..
- Body request (có thể có hoặc không): Cho phép client gửi đến yêu cầu bổ sung cần server thực hiện như: tạo mới hoặc cập nhật dữ liệu mà không thể truyền trên Header Parameters..
- URL nguồn(URL chung cho tất cả Api trong hệ thống).
- Parameter.
- Request header: Request header giúp client có thể gửi yêu cầu lên server. Mỗi yêu cầu sẽ kèm theo các thông số, và các thông số đó được gọi là *Header Parameters*.

HTTP Response

HTTP Response là gì?

HTTP Response



Sau khi nhận và phân tích thông điệp request, server sẽ phản hồi bằng một thông điệp HTTP response.

Các thành phần của HTTP Response:

- Status-line.
- Không có hoặc có thêm các thành phần header (General | Response | Entity) và theo sau là các ký hiệu CRLF(Carriage Return Line Feed – báo hiệu trở về đầu dòng tiếp theo).
- Một dòng trống (VD một dòng không có nội dung gì và kết thúc bởi CRLF) dùng để báo hiệu kết thúc các thành phần header.
- Có hoặc không có phần nội dung của thông điệp (message-body).

Status-Line, HTTP Version



Một status-line (dòng trạng thái) bao gồm thông tin về phiên bản của giao thức theo sau một con số thể hiện mã của trạng thái và các nội dung văn bản liên quan. Các thành phần được phân tách bởi dấu cách (space).

Một server hỗ trợ HTTP phiên bản 1.1 sẽ trả về thông tin phiên bản như sau:

```
HTTP-Version = HTTP/1.1
```

Status Code



Đây là thành phần quan trọng trong HTTP Response.

- *1xx: Informational* - Số này có nghĩa là request đã được nhận và đang được xử lý.
- *2xx: Success* - Số này báo hiệu sự thành công, hành động đã được nhận, được hiểu và chấp nhận.
- *3xx: Redirection* - Số này báo hiệu rằng phải thực hiện thêm hành động để hoàn tất request.
- *4xx: Client Error* - Số này báo hiệu request chứa các sai sót về cú pháp hoặc không thể thực thi.
- *5xx: Server Error* - Số này có nghĩa là server bị lỗi, nó không thực hiện được request mặc dù đó là request hợp lệ.

Response Header



Các thành phần của response-header cho phép server bổ sung các thông tin bổ sung response khi không thể đưa chúng vào Status-Line.

Ví dụ HTTP Response



HTTP/1.1 200 OK

Date: Mon, 27 Jul 2009 12:28:53 GMT

Server: Apache/2.2.14 (Win32)

Last-Modified: Wed, 22 Jul 2009 19:15:56 GMT

Content-Length: 88

Content-Type: text/html

Connection: Closed

<html>

<body>

<h1>Hello, World!</h1>

</body>

</html>

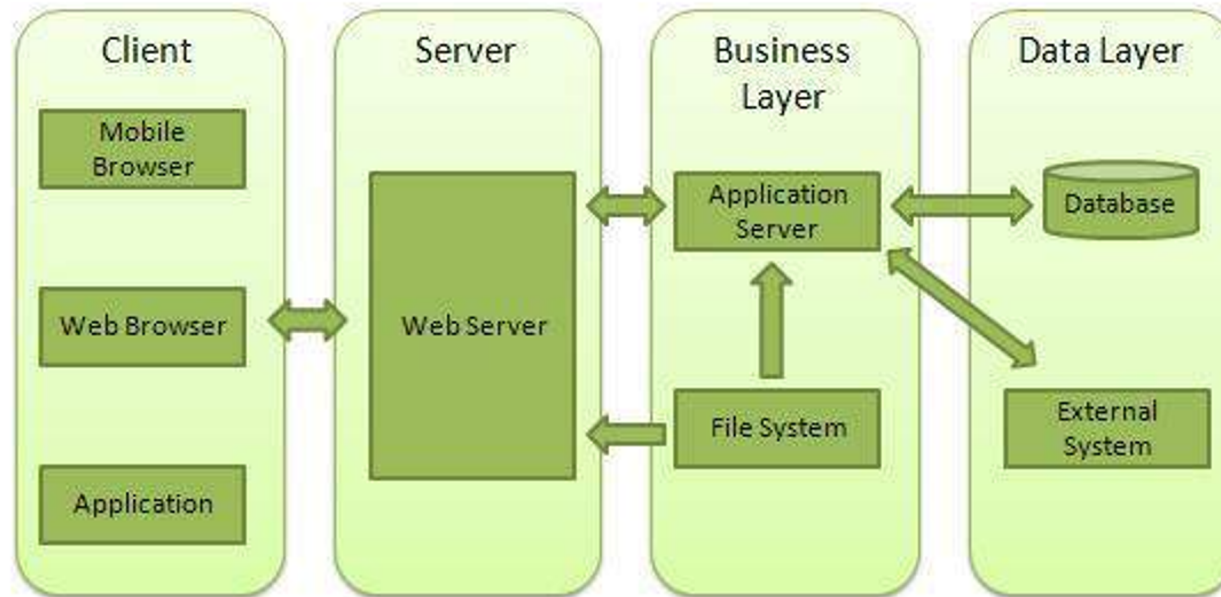


Ứng dụng server trong NodeJS.

Web Server là gì?



- Web Server là một ứng dụng phần mềm có thể xử lý các HTTP request được gửi bởi HTTP Client (ví dụ: trình duyệt web) và trả về một trang web trong phản hồi tới Client.
- Một ứng dụng web thường được chia thành 4 lớp như sau:



-
- Client - Lớp này bao gồm các trình duyệt web và các ứng dụng có thể tạo ra các HTTP request đến Web Server.
 - Server - Lớp này bao gồm các Web Server có thể can thiệp các request được tạo bởi Client và trả về các phản hồi (response).
 - Business - Lớp này bao gồm ứng dụng trên Server có thể được tận dụng bởi các Web Server để thực hiện các tiến trình xử lý cần thiết. Lớp này tương tác với lớp Data qua các chương trình bên ngoài.
 - Data - Lớp này bao gồm các Database và bất kì các nguồn dữ liệu nào.



Module HTTP trong NodeJS

- HTTP có thể hiểu đơn giản là module cho phép Node.js truyền dữ liệu qua giao thức http, cụ thể hơn chức năng của nó là tạo một máy chủ HTTP lắng nghe các cổng kết nối trên đó và trả lời lại cho máy khách (client).
- HTTP module là core module (được tích hợp sẵn khi cài đặt Node.js).
- Khai báo HTTP module:

```
const http = require('http');
```

- Phương thức tạo server của http - `createServer()`: tạo một máy chủ HTTP.

Ví dụ



```
const http = require('http');
const server = http.createServer((req, res) => {
  res.write('<h1> Hi, thank you for visit  
https://codegym.vn</h1><hr>');
  res.end();
});
server.listen(8080, '127.0.0.1');
```



Demo

Ví dụ về Web server đơn giản bằng Module HTTP.