HTTP

1. Tổng quan
   1. HTTP là gì?

HTTP là một giao thức cấp độ ứng dụng cho các hệ thống thông tin phân phối, cộng tác, đa phương tiện. HTTP là một giao thức giao tiếp trên cơ sở TCP/IP, mà được sử dụng để phân phối dữ liệu (các tệp HTML, các file ảnh, …) trên WWW. Cổng mặc định là TCP 80, những các cổng khác cũng có thể được sử dụng.

* 1. Đặc trưng cơ bản của HTTP
  + **HTTP là giao thức connectionless (kết nối không liên tục):** Client của HTTP, ví dụ: một trình duyệt khởi tạo một yêu cầu HTTP và sau đó một yêu cầu được tạo ra, Client ngắt kết nối từ Server và đợi cho một phản hồi. Server xử lý yêu cầu và thiết lập lại sự kết nối với Client để gửi phản hồi trở lại.
  + **HTTP là một phương tiện độc lập:** Nó nghĩa là, bất kỳ loại dữ liệu nào cũng có thể được gửi bởi HTTP miễn là Server và Client biết cách để kiểm soát nội dung dữ liệu. Nó được yêu cầu cho Client cũng như Server để xác định kiểu nội dung bởi sử dụng kiểu MIME thích hợp.
  + **HTTP là stateless:** Như đã được đề cập ở trên, HTTP là connectionless và nó một kết quả trực tiếp là HTTP trở thành một giao thức Stateless. Server và Client biết về nhau chi trong một yêu cầu hiện tại. Sau đó, cả hai chúng nó quên tất cả về nhau. Do bản chất của giao thức, cả Client và các trình duyệt có thể giữ lại thông tin giữa các yêu cầu khác nhau giữa các trang web.
  1. Cấu trúc cơ bản của HTTP



Giao thức HTTP là một giao thức Yêu cầu/Phản hồi dựa trên cấu trúc Client/Server, nơi mà các trình duyệt web, các thiết bị tìm kiếm,… hoạt động như các Client, và các Server web hoạt động như một Server.

* + - * Client : Clieant gửi một yêu cầu tới server theo mẫu của một phương thức yêu cầu, URI, và phiên bản giao thức, thược theo bởi một thông báo mime chứa bộ chỉnh yêu cầu, thông tin Client, và nội dung đối tượng có thể qua một kết nối TCP/IP.
      * Server: phản hồi với một dòng trạng thái bao gồm phiên bản giao thức của thông bao và một code thành công hoặc lỗi, theo sau bởi một thông báo MIME chứ thông tin server, thông tin thực thể đa phương tiện và nội dung đối tượng cụ thể.

1. Parameter ( Tham số )
2. Phiên bản HTTP ( HTTP version)

HTTP sử dụng sơ đồ đánh số **<major>.<minor>**  để chỉ phiên bản giao thức.

Cú pháp chung cho việc xác định số phiên bản HTTP:

HTTP-Version = "HTTP" "/" 1\*DIGIT "." 1\*DIGIT

Ví dụ: HTTP/1.0

1. Uniform Resource Identifiers - Bộ nhận diện nguồn tài nguyên đồng nhất

URI là một chuỗi được định dạng, nhạy cảm với chữ hoa-thường (case-insentive) theo một cách đơn giản chứa tên, vị trí,.. để xác định một nguồn. Cú pháp chung của Uri được sử dụng cho HTTP như sau:

URI = "http:" "//" host [ ":" port ] [ abs\_path [ "?" query ]]

ở đây các thông số là:

* + - * Port: Trống hoặc không được cung cấp thì mặc định là 80
      * Abs\_path: trống là tương đương với “/”
      * Các ký tự khác trong bộ thiết lập reserved và unsafe là tương đương với mã hóa “%” Hex Hex”.

Ví dụ: 3 uri này tương đương nhau

http://abc.com:80/~smith/home.html

http://ABC.com/%7Esmith/home.html

http://ABC.com:/%7esmith/home.html

1. Time format ( định dạng thời gian)

Tất cả các nhãn Ngày/Thời gian HTTP **Phải** được biểu diễn trong Greenwich Mean Time (GMT), không có sự ngoại trừ. Các ứng dụng HTTP được cho phép để sử dụng 3 nhãn đại diện Ngày/Thời gian sau:

Sun, 06 Nov 1994 08:49:37 GMT ; RFC 822, updated by RFC 1123

Sunday, 06-Nov-94 08:49:37 GMT ; RFC 850, obsoleted by RFC 1036

Sun Nov 6 08:49:37 1994 ; ANSI C's asctime() format

1. Các bộ ký tự

Chúng ta sử dụng các bộ ký tự để xác định các thiết lập ký tự mà Client ưa thích. Để liệt kê nhiều bộ ra cùng thì ta có thể dùng dấu phảy. Nếu giá trị không được xác định thì mặc định là US-ASII.

Ví dụ: US-ASII, ISO-8859-1, ISO-8859-7

1. Mã hóa nội dung

Một giá trị mã hóa nội dung chỉ rằng một thuật toán mã hóa đã được sử dụng để mã hóa nội dung trước khi truyền nó tới mạng. Mã hóa nội dung được sử dụng lần đầu để cho phép một tài liệu để được nén hoặc ngoài ra được truyền tải mà không thất lạc sự nhận diện.

Tất cả các giá trị mã hóa nội dung là không phân biệt kiểu chữ (case-insensitive). HTTP/1.1 sử dụng các giá trị mã hóa nội dung trong các trường Accept-Encoding và Content-Encoding Heade.

Ví dụ:

Accept-encoding: gzip

or

Accept-encoding: compress

or

Accept-encoding: deflate

1. Media types ( các kiểu phương tiện)

HTTP sử dụng các Kiểu phương tiện Internet trong các trường **Content-Type** và **Accept** để cung cấp dữ liệu mở và có thể mở rộng. Tất cả các giá trị kiểu phương tiện được đăng ký với IANA (Internet Assigned Number Authority). Cú pháp chung để xác định kiểu phương tiện như sau:

media-type = type "/" subtype \*( ";" parameter )

ví dụ: accept: image/gif

1. Language-tag (thẻ ngôn ngữ)

HTTP sử dụng các thử ngôn ngữ trong các trường hợp Accept-Language và Content-Language. Một thẻ ngôn ngữ gồm nhiều thành phần:

language-tag = primary-tag \*( "-" subtag )

Các khoảng trắng không được cho phespt trong thẻ.

Ví dụ: en, en-US, en-cockney, i-cherokee, x-pig-latin  
 Trong đó “en” là viết tắt cho ngôn ngữ trong ISO-639 và hai ký tự đầu tiên trong thẻ subtag là mã quốc gia

1. Message trong HTTP

HTTP được xây dựng trên cơ sở mô hình cấu trúc Client-Server và giao thức Stateless các Yêu cầu/Phản hồi mà điều hành bởi việc trao đổi các thông báo (Message) dọc theo một kết nối TCP/IP.

* + - * HTTP sử dụng URI để nhận một nguồn để cho và để thiết lập một kết nối. một khi một kết nối được thiết lập, các thông báo http được truyền theo dạng tương tự Internet Mail [RFC5322] và MIME [RFC2045].
      * Yêu cầu từ Client tới server và phản hồi về sẽ theo định dạng như sau:

HTTP-message = <Request> | <Response> ; HTTP/1.1 messages

Các phản hồi sử dụng một định dạng thông báo chung của RFC 822 bao gồm 4 mục:

* + Một dòng đầu tiên
  + Không hoặc nhiều trường Header theo sau bởi CRLF
  + Một dòng trống ( ví dụ: một dòng mà không có gì trước CRLF), chỉ phần cuối của trường Header.
  + Một thân thông báo tùy ý

1. Dòng đầu thông báo (start-line)

Một dòng đầu sẽ có cú pháp chung như sau:

start-line = Request-Line | Status-Line

Ví dụ yêu cầu và phản hồi:

GET /hello.jsp HTTP/1.1 (This is Request-Line sent by the client)

HTTP/1.1 200 OK (This is Status-Line sent by the server)

1. Các trường Header

Các header cung cấp thông tin được yêu cầu hoặc phản hồi, hoặc đối tượng trong thông báo. Có 4 kiểu của Header trong thông báo:

* + General-Header: có khả năng ứng dụng chung cho tất cả các thông báo.
  + Request-Header: chỉ có khả năn áp dụng cho các thông báo yêu cầu.
  + Response-Header: chỉ áp dụng cho các thông báo phản hồi.
  + Entity-Header: xác định thông tin về thân thực thể.

Tất cả các header đều theo định dạng chung như sau:

message-header = field-name ":" [ field-value ]

ví dụ về các trường Header đa dạng:

User-Agent: curl/7.16.3 libcurl/7.16.3 OpenSSL/0.9.7l zlib/1.2.3

Host: www.example.com

Accept-Language: en, mi

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

Server: Apache

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

ETag: "34aa387-d-1568eb00"

Accept-Ranges: bytes

Content-Length: 51

Vary: Accept-Encoding

Content-Type: text/plain

1. Phần thông báo

Phần thân thông báo có thể mang phần thân được liên kết với yêu cầu hoặc phản hồi. Nếu phần thông được liên kết thì sau đó thường các dòng Centent-Type và Content-Length xác định bản chất của phần thân được liên kết. Một phần thông báo là phần mang dữ liệu được yêu cầu ( dư liệu mẫu và được tải lên,…) và dữ liệu phản hồi từ server( ảnh, file,…)

Ví dụ:

<html>

<body>

<h1>Hello, this’s message!</h1>

</body>

</html>

1. Header http

Có 4 loại header đã liệt kê bên trên.

1. General Header
   * Cache-Control

Được sử dụng để xác định các đường dãn mà phải tuân theo bởi tất cả các hệ thống bộ nhớ ẩn.

Cache-Control : cache-request-directive|cache-response-directive

Có thể dùng để xác định các tham số cho bộ nhớ ẩn hoặc yêu cầu tài liệu từ bộ nhớ ẩn. Tiếp theo ta sẽ liệt kê các chỉ dẫn bộ nhớ ẩn quan trọng:

|  |  |
| --- | --- |
| **STT** | **Chỉ dẫn yêu cầu bộ nhớ ẩn và miêu tả** |
| 1 | **no-cache**  Một bộ nhớ ẩn phải không sử dụng phản hồi để làm thỏa mãn một yêu cầu theo sau mà không tái xác nhận thành công với Server ban đầu. |
| 2 | **no-store**  Bộ nhớ ẩn không nên lưu giữ bất cứ thứ gì về yêu cầu Client hoặc phản hồ Server. |
| 3 | **max-age = giây (s)**  Chỉ ra rằng Client đang muốn chấp nhận một phản hồi mà thời gian của nó không lớn hơn thời gian đã xác định bằng giây (s). |
| 4 | **max-stale [ tính bằng giây ]**  Chỉ ra rằng Client đang muốn chấp nhận một phản hồi mà đã vượt thời gian mãn hạn. Nếu số giây được cung cấp, nó phải không là hết hạn bởi nhiều hơn thời gian đó. |
| 5 | **min-fresh = giây**  Chỉ ra rằng Client đang muốn chấp nhận một phản hồi mà thời gian sống khỏe của nó là không ít hơn tuổi hiện tại của nó cộng với thời gian đã xác định bằng giây. |
| 6 | **no-transform**  Không chuyển đổi phần thân đối tượng. |
| 7 | **only-if-cached**  Không lấy dữ liệu mới. Bộ nhớ ẩn có thể gửi một tài liệu chỉ khi nó ở trong bộ nhớ ẩn, và không nên liên hệ với Server ban đầu để xem xét nếu một bản sao mới hơn tồn tại. |

* + Connection

Cho phép người gửi xác định các chức năng mà được mong ước cho kết nối cụ thể.

Connection : "Connection"

“Close” sẽ đóng kết nối sau khi hoàn thành phản hồi.

Connection: close

“keep-alive” sử dụng các kết nối liên tục, không tự động đóng sau khi hoàn thành một giao dịch.

Connection: keep-alive

* + Date : tất cả phải được biểu diễn trong GMT
  + Pragma

Dùng để chỉ dẫn cụ thể để thực hiện mà có thể áp dụng tới bất kì người nhận nào trong chuỗi yêu cầu.

* + Trailer : thiết lập các header được mã hóa và truyền tải được đóng gói

Trailer : field-name

Các trường Header thông báo được liệt kê trong Trailer phải không bao gồm các trường Header sau:

Transfer-Encoding

Content-Length

Trailer

* + Tranfer-encoding

Là kiểu truyền tải được áp dụng tới thân thông báo để cho việc truyền tải một cách an toàn từ người gửi đến người nhận.

Transfer-Encoding: chunked

* + Upgrade

Cho phép Client xác định những giao thức giao tiếp thêm vào mà nó hỗ trợ và sẽ sử dụng nếu server tìm thấy thích hợp để chuyển đổi giao thức.

Upgrade: HTTP/2.0, SHTTP/1.3, IRC/6.9, RTA/x11

Upgrade được chờ đợi để cung cấp một kỹ thuật đơn giản cho truyền tải từ HTTP/1.1 tới một số giao thức không tương hợp

* + Via

Sử dụng bởi gateway và các trạm ủy nhiệm để chỉ ra các giao thức trung gian với người nhận. HTTP/1.0 User agent tới một trạm ủy nhiệm nội bộ được đặt tên mã "fred", mà sử dụng HTTP/1.1 để chuyển tiếp yêu cầu tới một trạm ủy nhiệm công cộng tại nowhere.com, mà hoàn thành yêu cầu bởi việc chuyển tiếp nó tới Server ban đầu tại www.ics.uci.edu. Yêu cầu được nhận bởi www.ics.uci.edu sẽ có trường Via như sau:

Via: 1.0 fred, 1.1 nowhere.com (Apache/1.1)

* + Warning

Được sử dụng để mang thông tin thêm về status hoặc truyền tải một thông báo không được phản ánh trong đó.

Warning : warn-code warn-agent warn-text warn-date

1. Request header
   * Accept

Sử dụng để xác định các kiểu phương tiện cụ thể nào được hiển thị dưới dạng MIME.

Accept: type/subtype [q=qvalue]

Các kiểu phương tiện đực liệt kê phân biệt nhau bởi các dấu phảy và giá trị q là chất lượng có thể chấp nhận có phạm vi từ 0 đến 1.

Accept: text/plain; q=0.5, text/html, text/x-dvi; q=0.8, text/x-c

Đoạn này có thể được biên dịch như text/html và text/x-c và là các kiểu phương tiện được ưa thích hơn nhưng nếu chúng không tồn tại, thì sau đó gửi đối tượng text/x-dvi , và nếu nó không tồn tại, gửi đối tượng text/plain.

* + Accept-charset

Chỉ các bộ thiết lập ký tự nào được chấp nhận cho phản hồi.

Accept-Charset: character\_set [q=qvalue]

Quy tắc của accept-charset cũng giống với accept bên trên. Đặc biệt nếu không có giá trị nào thì để \* là bộ ký tự nào cũng sẽ được chấp nhận.

* + Accept-Encoding

Cũng tương tự như Accept, nhưng hạn chế mã hóa nội dung là có thể chấp nhận trong phản hồi, cú pháp:

Accept-Encoding: encoding types

* + Accept-Language

Cũng tương tự như Accept, nhưng hạn chế bộ thiết lập của các ngôn ngữ tự nhiên là được ưa thích hơn khi một phản hồi tới yê càu. Cú pháp:

Accept-Language: language [q=qvalue]

* + Authorization

Bao gồm các sự uy nhiệm mà chứa thông tin ủy quyền của một user agent cho phạm vi nguồn đang yêu cầu. Cú pháp:

Authorization : credentials

* + Cookie

Giá trị của value chứa một cặp tên, giá trị của thông tin được lưu giữ cho URL. Cú pháp:

Cookie: name=value

Nhiều cookie có thể được xác định phân biệt nhau bởi các dấu chấm phảy ";" như sau:

Cookie: name1=value1;name2=value2;name3=value3

* + Expect

Expect được sử dụng để chỉ ra thiết lập cụ thể của các hành vi server được yêu cầu bởi client. Cú pháp:

Expect : 100-continue | expectation-extension

* + From

Chứa địa chỉ email cho người sử dụng kiểm soát user agent. Cú pháp:

From: [webmaster@w3.org](mailto:webmaster@w3.org)

* + Host

Được sử dụng để xác định host và port của nguồn được yêu cầu. Cú pháp:

Host : "Host" ":" host [ ":" port ] ;

Một host không có port thì mặc định là 80.

* + If-Match

Được sử dụng trong một method để làm cho có điều kiện. Header này yêu cầu server để biểu diễn method được yêu cầu chỉ khi giá tị được cung cấp trong thẻ này kết nối với các thẻ đối tượng.

If-Match : entity-tag

Dấu \* kết nối với bất kì đối tượng nào, và sự truyền tải tiếp tục khi đối tượng. Ví dụ:

If-Match: "xyzzy"

If-Match: "xyzzy", "r2d2xxxx", "c3piozzzz"

If-Match: \*

* + If-modified-since

Sử dụng với một method để làm cho nó có điều kiện. Nếu URL được yêu cầu không được chỉnh sửa từ thời gian đã được xác định trong trường này, một đối tượng sẽ không được trả lại từ server; thay vào đó, một phản hồi 304 ( không được chỉnh sửa) sẽ được trả lại mà không có bất cứ phần thân thông báo nào. Chú pháp:

If-Modified-Since : HTTP-date

Một ví dụ của trường là:

If-Modified-Since: Sat, 29 Oct 1994 19:43:31 GMT

Nếu không có thẻ đối tượng nào kết nối hoặc \* được cung cấp và không đối tượng hiện tại nào tồn tại. Server không được trình bày method được yêu cầu và trả lại lỗi 412 (điều kiện trước thất bại).

* + If-none-match

Sử dụng một method để làm cho nó có điều kiện. Trường này yêu cầu server trình bày method được yêu cầu chỉ khi một trong số giá trị đã cho của thẻ này kết nối với các đối tượng. Cú pháp:

If-None-Match : entity-tag

* + If-range

Được sử dụng với một GET có điều kiện để yêu cầu chỉ một phàn của đối tượng mà đang bị thất lạc, nếu nó không được thay đổi toàn bộ đối tượng nếu nó được thay đổi. Cú pháp:

If-Range : entity-tag | HTTP-date

* + If-unmodified-since

Được sử dụng với một method để làm cho nó có điều kiện. Cú pháp:

If-Unmodified-Since : HTTP-date

Nếu yêu cầu có kết quả là bất cứ gì khách ngoài một trạng thái 2xx hoặc 4xx, thì trường if-unmodified-since sẽ bỏ qua.

* + Max-Forwards

Cung cấp phương thức TRACE và OPTIONS để giới hạn số các proxy hoặc gateway mà có thể chuyển tiếp yêu cầu server kế tiếp. Cú pháp :

Max-Forwards : n

------------------------

1. Request

Một client gửi một yêu cầu HTTP tới một Server trong mẫu một thông báo yêu cầu bao gồm định dạng như sau:

* + Request-Line
  + Trống hoặc Header (General | request | entity) follow by CRLF
  + Trống hoặc chỉ phần kết thúc của Header.
  + Thân thông báo tùy ý.

1. Request-Line ( Dòng yêu cầu)
   * + - Bắt đầu bằng một phương thức (method), tiếp tho là request-uri, kết thúc bằng CRLF, các yếu tố được phân biệt bởi dấu cách.

Request-Line = Method Request-URI HTTP-Version CRLF

1. Phương thức yêu cầu

Phương thức yêu cầu chỉ phương thức để được thực hiện trên nguồn được nhận diện bởi Request-URI đã cung cấp. Method là case-intensive và luôn được viết hoa.

|  |  |
| --- | --- |
| STT | Phương thức vào miêu tả |
| 1 | GET  GET được sử dụng để lấy thông tin từ Server đã cung cáp bởi Request-URI. Các yêu cầu sử dụng GET nên chỉ nhận dữ liệu nên không ảnh hưởng gì tới dữ liệu |
| 2 | HEAD  Tương tự như GET, nhưng có truyền tải status và header. |
| 3 | POST  Một yêu cầu POST được sử dụng để gửi dữ liệu tới server (thông tin khách hàng, file,…) bởi dử dụng HTML. |
| 4 | PUT  Thay đổi tất cả các dữ liệu hiện tại với nội dung tải lên mới (update) |
| 5 | DELETE  Xóa bỏ các dữ liệu đã có thông qua thông tin được truyền trên URI. |
| 6 | CONNECT  Giao tiếp với tài nguyên được yêu cầu được xác định trong Request-URI |
| 7 | OPTIONS  Miêu tả các chức năng giao tiếp. |
| 8 | TRACE  Trình bày một vòng lặp kiểm tra thông báo song song với path tới nguồn mục tiêu. |

1. Request-uri

Là Uniform Source Identifier (URI) và xác định nguồn để áp dụng yêu cầu.  
mẫu sử dụng xác định một URI:

Request-URI = "\*" | absoluteURI | abs\_path | authority

|  |  |
| --- | --- |
| STT | Phương thức và Miêu tả |
| 1 | Một dấu \* có nghĩa là yêu cầu không áp dụng cho một tài nguyên cụ thể mà cho chính máy chủ và chỉ được phép khi phương thức được sử dụng không nhất thiết phải áp dụng cho một tài nguyên.  Ví dụ: **OPTIONS \* HTTP/1.1** |
| 2 | absoluteURI được sử dụng khi một yêu cầu đang được tạo ra cho một proxy. Proxy được yêu cầu chuyển tiếp yêu cầu hoặc dịch vụ hợp lệ và trả lại phản hồi.  Ví dụ: GET http://www.w3.org/pub/WWW/TheProject.html HTTP/1.1 |
| 3 | Mẫu phổ biến nhất của request-URI được sử dụng để xác định một nguồn trên một server hoặc gateway ban đầu. Ví dụ một Client mong muốn lấy được một tài nguyên tực tiếp từ server ban đầu sẽ tạo một kết nối TCP tới port 80 của host [www.w3.org](http://www.w3.org) và gửi các dòng sau:  **GET /pub/WWW/TheProject.html HTTP/1.1**  **Host: www.w3.org**  Lưu ý: đường truyền tuyệt đối không thể là trống; nếu không thì trình bày tỏng URI ban đầu, phải được cung cấp là “/” (server root) |

1. Các trường Header yêu cầu.

Các Request-Header cho phép Client cho phép Client chuyển thông tin về yêu cầu chính client về server. Nhữn trường hợp này hoạt động như cac bộ chỉnh sửa request.

Các danh sách Request-Header quan trọng:

* Accept-Charset
* Accept-Encoding
* Accept-Language
* Authorization
* Expect
* From
* Host
* If-Match
* If-Modified-Since
* If-None-Match
* If-Range
* If-Unmodified-Since
* Max-Forwards
* Proxy-Authorization
* Range
* Referer
* TE
* User-Agent

1. Response

Sau khi nhận và dịch một số thông báo yêu cầu, một server gửi tín hiệu phản hồi với một thông báo:

1. Status line

Một dòng trạng thái bao gồm phiên bản giao thức được theo sau bỏi trạng thái và văn bản được liên kết với nó.

Status-Line = HTTP-Version SP Status-Code SP Reason-Phrase CRLF

1. Phiên bản HTTP

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

HTTP-version = HTTP/1.1

1. Status code

Là một số nguyên có 3 ký tự, trong đó ký tự đầu tiên là loại phản hồi.

Có 5 giá trị của ký tự đầu tiên:

|  |  |
| --- | --- |
| STT | Mã và miêu tả |
| 1 | 1xx: thông tin  Nó nghĩa là yêu cầu đã được nhận và tiến trình đang tiếp tục. |
| 2 | 2xx: thành công  Nó nghĩa là hoạt động đã được nhận, được hiểu và chấp nhận thành công. |
| 3 | 3xx: điều hướng  Nó nghĩa là hoạt động phải được thực hiện để hoàn thành yêu cầu. |
| 4 | 4xx: lỗi client  Nó nghĩa là yêu cầu chứa cú pháp không chính xác hoặc không được thực hiện yêu cầu. |
| 5 | 5xx: lỗi server  Nó nghĩa là server thất bại với việc thực hiện một yêu cầu nhìn như có vẻ khả thi. |

Chi tiết các mã:

**1xx**: thông tin

|  |  |
| --- | --- |
| **Thông báo** | **Miêu tả** |
| **100** Continue | Chỉ một phần của yêu cầu được nhận bởi server, nhưng miễn là nó không bị loại bỏ thì Client tiếp tục được yêu cầu |
| **101** Switching protocols | Server chuyển đổi giao thức |

**2xx**: Thành công

|  |  |
| --- | --- |
| **Thông báo** | **Miêu tả** |
| **200** OK | Yêu cầu OK |
| **201** Created | Yêu cầu hoàn thành, nguồn mới được tạo. |
| **202** Accepted | Yêu cầu được chấp nhận cho xử lý, nhưng việc xử lý chưa hoàn thành. |
| **203** Non-authoritative information | Thông tin trong đối tượng Header là từ một bản sao nội bộ hoặc bên thứ 3, không từ server ban dầu. |
| **204** No content | Một status code và một header được cung cấp trong phản hồi, nhưng không có phần thân đối tượng trong sự phản hồi. |
| **205** Reset Content | Trình duyệt nên dọn sạch mẫu được sử dụng cho việc truyền tải này bởi một dữ liệu đầu vào tăng thêm. |
| **206** partial content | Server đang trả lại dữ liệu cục bộ của kích cỡ được yêu cầu. Được sử dụng trong phản hồi tới một yêu cầu xác định một range header. Server phải xác định dãy được bao gồm trong phản hồi với Content-range |

**3xx**: Điều hướng

|  |  |
| --- | --- |
| **Thông báo** | **Miêu tả** |
| **300** Multiple Choices | Một danh sách các link. Người sử dụng có thể chọn một link và tới vị trí đó. Tối đa 5 địa chỉ. |
| **301** Moved Permanently | Trang được yêu cầu đã di chuyển tới một URL mới. |
| **302** Found | Trang được yêu cầu đã di chuyển tạm thời tới một URL mới. |
| **303** See Other | Trang được yêu cầu có thể được tìm ở dưới một URL khác. |
| **304** Not Modified | Đây là mã phản hồi tới một If-Modified-Since hoặc If-None-Match header, nơi mà URL không được chỉnh sửa từ ngày cụ thể. |
| **305** Use Proxy | URL được yêu cầu phải được truy cập thông qua một sự ủy quyền được đề cập trong Location Header. |
| **306** Unused | Mã này được sử dụng trong một phiên bản trước. Nó không còn được sử dụng nữa, nhưng mã này được lưu giữ. |
| **307** Temporary Redirect | Trang được yêu cầu đã di chuyển tạm thời tới một URL mới. |

**4xx**: Lỗi Client

|  |  |
| --- | --- |
| **Thông báo** | **Miêu tả** |
| **400** Bad Request | Server không hiểu yêu cầu. |
| **401** Unauthorized | Trang được yêu cầu cần một tên sử dụng và một mật khẩu. |
| **402** Payment Required | Bạn không thể sử dụng mã này nữa.. |
| **403** Forbidden | Sự truy cập tới trang được yêu cầu bị cấm. |
| **404** Not Found | Server không thể tìm thấy trang được yêu cầu. |
| **405** Method Not Allowed | Phương thức được xác định trong yêu cầu là không được cho phép. |
| **406** Not Acceptable | Server chỉ có thể tạo một phản hồi mà không được chấp nhận bởi Client. |
| **407** Proxy Authentication Required | Bạn phải xác nhận với một Server ủy quền trước khi yêu cầu này được phục vụ. |
| **408** Request Timeout | Yêu cầu tốn thời gian dài hơn thời gian Server được chuẩn bị để đợi. |
| **409** Conflict | Yêu cầu không thể được hoàn thành bởi vì sự xung đột. |
| **410** Gone | Trang được yêu cầu không có sẵn nữa. |
| **411** Length Required | Content-Length không được xác định rõ. Server sẽ không chấp nhận yêu cầu mà không có nó. |
| **412** Precondition Failed | Điều kiện trước được cung cấp trong yêu cầu được tính toán là sai bởi Server. |
| **413** Request Entity Too Large | Server sẽ không chấp nhận yêu cầu, bởi vì đối tượng yêu cầu là quá rộng. |
| **414** Request-url Too Long | Server sẽ không chấp nhận yêu cầu, bởi vì URL là quá dài. Xảy ra khi bạn chuyển một yêu cầu "port" tới một yêu cầu "get" với thông tin quá dài. |
| **415** Unsupported Media Type | Server sẽ không chấp nhận yêu cầu, bởi vì kiểu phương tiện không được hỗ trợ. |
| **416** Requested Range Not Satisfiable | Dãy byte được yêu cầu là không có sẵn và bên ngoài giới hạn. |
| **417** Expectation Failed | Khả năng được cung cấp trong một trường Expect không thể được kết nối bởi Server. |

**5xx**: Lỗi server

|  |  |
| --- | --- |
| **Thông báo** | **Miêu tả** |
| **500** Internal Server Error | Yêu cầu không được hoàn thành. Server bắt gặp một điều kiện không được mong đợi. |
| **501** Not Implemented | Yêu cầu không được hoàn thành. Server không hỗ trợ tính năng được yêu cầu. |
| **502** Bad Gateway | Yêu cầu không được hoàn thành. Server nhận một phản hồi không có hiệu lực từ Server ở thượng nguồn. |
| **503** Service Unavailable | Yêu cầu không được hoàn thành. Server tạm thời đang quá tải hoặc down. |
| **504** Gateway Timeout | Gateway bị trễ. |
| **505** HTTP Version Not Supported | Server không hỗ trợ phiên bản "giao thức HTTP". |

1. Header phản hồi

Các header phản hồi cho phép server truyền thông tin thêm về phản hồi mà không thể đặt được trong status-line. Những header này cung cấp htoong tin về server và về truy cập từ xa tới nguồn được xác định bởi request-uri.

* Accept-Ranges
* Age
* ETag
* Location
* Proxy-Authenticate
* Retry-After
* Server
* Vary
* WWW-Authenticate