

Yêu cầu (Request) trong HTTP

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

- Một dòng yêu cầu

- Không hoặc nhiều hơn trường Header (`GeneralRequestEntity`) được theo sau bởi CRLF.

- Một dòng trống (ví dụ: một dòng không có gì đằng trước CRLF) chỉ phần kết thúc của trường Header.

- Một phần thân thông báo tùy ý

Các phần dưới giải thích cách sử dụng của mỗi đối tượng trong thông báo yêu cầu HTTP.

Dòng Yêu cầu

Dòng Yêu cầu bắt đầu với một thủ tục method, được theo sau bởi một Request-URI và phiên bản giao thức, và kết thúc với CRLF. Các yếu tố được phân biệt riêng rẽ bởi các ký tự khoảng trống SP.

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

Dưới đây thảo luận về mỗi phần được đề cập trong Dòng Yêu cầu.

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 nên luôn luôn được đề cập trong chữ hoa. Bảng dưới đây liệt kê tất cả các method được hỗ trợ trong HTTP/1.1.

STT	Phương thức và Miêu tả
1	GET GET được sử dụng để lấy lại thông tin từ Server đã cung cấp bởi sử dụng một URI đã cung cấp. Các yêu cầu sử dụng GET nên chỉ nhận dữ liệu và nên không có ảnh hưởng gì tới dữ liệu.
2	HEAD Tương tự như GET, nhưng nó truyền tải dòng trạng thái và khu vực Header.
3	POST Một yêu cầu POST được sử dụng để gửi dữ liệu tới Server, ví dụ, thông tin khách hàng, file tải lên, ..., bởi sử dụng các mẫu HTML.
4	PUT Thay đổi tất cả các đại diện hiện tại của nguồn mục tiêu với nội dung được tải lên.
5	DELETE Gỡ bỏ tất cả các đại diện hiện tại của nguồn mục tiêu bởi URI.
6	CONNECT Thiết lập một tunnel tới Server được xác định bởi URI đã cung cấp.
7	OPTIONS Miêu tả các chức năng giao tiếp cho nguồn mục tiêu.
8	TRACE

Trình bày một vòng lặp kiểm tra thông báo song song với path tới nguồn mục tiêu.

Request-URI

Request-URI là một Bộ nhận diện Nguồn Đồng nhất (URI) và xác định nguồn mà áp dụng yêu cầu. Dưới đây là các mẫu thường được sử dụng để xác định một URI:

```
Request-URI = "*" | absoluteURI | abs_path | authority
```

STT	Phương thức và Miêu tả
1	<p>Một dấu * được sử dụng khi một yêu cầu HTTP không áp dụng tới một nguồn cụ thể, nhưng tới chính Server đó, và chỉ được cho phép khi phương thức được sử dụng không cần thiết áp dụng tới một nguồn. Ví dụ:</p> <p>OPTIONS * HTTP/1.1</p>
2	<p>absoluteURI được sử dụng khi một yêu cầu HTTP đang được tạo ra cho một sự ủy nhiệm. Sự ủy nhiệm được yêu cầu chuyển tới yêu cầu hoặc dịch vụ từ một cache hiệu lực, và trả lại phản hồi. Ví dụ:</p> <p>GET http://www.w3.org/pub/WWW/TheProject.html HTTP/1.1</p>
3	<p>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 nguồn một cách trực tiếp từ Server ban đầu sẽ tạo một kết nối TCP tới port 80 của host www.w3.org và gửi các dòng sau:</p> <p>GET /pub/WWW/TheProject.html HTTP/1.1</p> <p>Host: www.w3.org</p> <p>Ghi chú rằng, đường truyền tuyệt đối không thể là trống rỗng; nếu không gì được trình bày trong URI ban đầu, nó Phải được cung cấp như là "/" (Server root).</p>

Các trường Header Yêu cầu

Chúng ta sẽ học General-Header và Entity-Header trong một chương riêng khi chúng ta sẽ học về các trường Header. Bây giờ, chúng ta xem các trường Header yêu cầu là gì:

Các trường Request-Header cho phép Client truyền thông tin thêm về yêu cầu, và về chính Client đó, tới Server. Những trường này hoạt động như các bộ chỉnh sửa yêu cầu. Dưới đây là một danh sách các trường Request-Header quan trọng mà có thể được sử dụng dựa trên sự yêu cầu:

- 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

Bạn có thể tự tạo các trường cho mình trong trường hợp bạn đang viết Client và Server cho riêng mình.

Các ví dụ của Thông báo Yêu cầu

Bây giờ chúng ta đặt tất cả những thứ đã học ở trên cùng với nhau để tạo một yêu cầu HTTP để chỉ thị trang **hello.htm** từ Server chạy trên tutorialspoint.com.

```
GET /hello.htm HTTP/1.1 User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT) Host: www.tutorialspoint.com Accept-Language: en-us Accept-Encoding: gzip, deflate Connection: Keep-Alive
```

Tại đây chúng ta không gửi bất cứ yêu cầu dữ liệu tới Server bởi vì chúng ta đang chỉ thị một trang thuần HTML từ Server. Kết nối là General-Header, và phần còn lại của Header là các Header yêu cầu. Ví dụ sau đây chỉ cách để gửi dữ liệu mẫu tới Server bởi sử dụng phần thân thông báo yêu cầu:

```
POST /cgi-bin/process.cgi HTTP/1.1 User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT) Host: www.tutorialspoint.com Content-Type: application/x-www-form-urlencoded Content-Length: length Accept-Language: en-us Accept-Encoding: gzip, deflate Connection: Keep-Alive  
licenseID=string&content=string&/paramsXML=string
```

Ở đây, URI được cung cấp */cgi-bin/process.cgi* sẽ được sử dụng để xử lý dữ liệu được truyền và theo đó, một phản hồi sẽ được trả lại. Ở đây **content-type** nói cho Server rằng dữ liệu được truyền là một dữ liệu mẫu web đơn giản và **length** sẽ là độ dài thực của dữ liệu đặt trong phần thân thông báo. Ví dụ sau chỉ cách bạn có thể truyền XML thuần tới Server của bạn.

```
POST /cgi-bin/process.cgi HTTP/1.1 User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT) Host: www.tutorialspoint.com Content-Type: text/xml; charset=utf-8 Content-Length: length Accept-Language: en-us Accept-Encoding: gzip, deflate Connection: Keep-Alive <?xml version="1.0" encoding="utf-8"?>  
<string xmlns="http://clearforest.com/">string</string>
```