

# 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.

Một Client là một chương trình (một trình duyệt hoặc bất kỳ Client) mà thiết lập một kết nối tới một Server cho mục đích gửi một hoặc nhiều thông báo yêu cầu HTTP. Một HTTP “Server” là một chương trình (hiểu theo cách chung là một Server web như Apache Server web hoặc Internet Information Services – IIS ...) mà chấp nhận các kết nối để phục vụ các yêu cầu HTTP bởi việc gửi các thông báo phản hồi HTTP.

HTTP sử dụng URI để nhận diện một nguồn đã cho và để thiết lập một kết nối. Một khi một kết nối được thiết lập, Các **Thông báo HTTP** được truyền trong một định dạng tương tự như được sử dụng trong thư điện tử Internet Mail [RFC5322] và MIME (Multipurpose Internet Mail Extensions) [RFC2045]. Các thông báo này bao gồm các **Yêu cầu** từ Client tới Server và các **Phản hồi** từ Server tới Client mà sẽ theo định dạng sau:

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

Các yêu cầu HTTP và các phản hồi HTTP sử dụng một định dạng thông báo chung của RFC 822 cho truyền tải dữ liệu được yêu cầu. Định dạng thông báo chung này 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

Trong các mục tiếp theo, chúng ta sẽ giải thích từng mục được sử dụng trong thông báo HTTP.

## Dòng đầu thông báo (start-line)

Một dòng đầu sẽ có cú pháp chung như sau:

```
start-line = Request-Line | Status-Line
```

Chúng ta sẽ bàn luận **Request-Line** và **Status-Line** trong khi thảo luận về các thông báo Yêu cầu HTTP và Phản hồi HTTP tương ứng. Bây giờ, chúng ta xem xét một số ví dụ về dòng bắt đầu trong trường hợp 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)
```

## Các trường Header

Các trường Header cung cấp thông tin được yêu cầu về yêu cầu hoặc phản hồi, hoặc về đối tượng được gửi trong thân thông báo. Có 4 kiểu của Header trong các thông báo HTTP:

- **Kiểu chung (General-Header):** Các trường Header này có khả năng ứng dụng chung cho cả các thông báo yêu cầu và phản hồi.
- **Kiểu yêu cầu (Request-Header):** Các trường Header này chỉ có khả năng áp dụng cho các thông báo yêu cầu.
- **Kiểu phản hồi (Response-Header):** Các trường Header này chỉ có khả năng áp dụng cho các thông báo phản hồi.
- **Kiểu thực thể (Entity-Header):** Các trường này xác định thông tin về thân-thực thể hoặc, nếu không có phần thân nào hiển thị, về nguồn được nhận diện bởi yêu cầu.

Tất cả các Header được đề cập ở trên theo một định dạng chung và mỗi một trường Header bao gồm một tên được theo sau bởi một dấu hai chấm (:) và giá trị trường như sau:

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

Dưới đây là 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
```

## Phần thân thông báo

Phần thân thông báo là tùy ý cho một thông báo HTTP nhưng nếu nó là có sẵn, thì khi đó nó được sử dụng để mang phần thân được liên kết với yêu cầu hoặc phản hồi. Nếu phần thân thực thể được liên kết, thì sau đó thường các dòng Content-Type và Content-Length xác định bản chất của phần thân được liên kết.

Một phần thân thông báo là phần mà mang dữ liệu yêu cầu HTTP thực sự (bao gồm dữ liệu mẫu và được tải lên,...) và dữ liệu phản hồi HTTP từ Server (bao gồm các file, ảnh, ...). Dưới đây là nội dung đơn giản của một phần thân thông báo:

```
<html> <body> <h1>Hello, World!</h1> </body> </html>
```

Hai chương tới chúng ta sẽ bàn về cách sử dụng của các khái niệm được giải thích để chuẩn bị Các yêu cầu HTTP và Các phản hồi HTTP.