

Phản hồi (Response) trong HTTP

Sau khi nhận và phiên dịch một thông báo yêu cầu, một Server gửi tín hiệu phản hồi với một thông báo phản hồi HTTP.

- Một dòng trạng thái (Status-Line)

- Không hoặc nhiều hơn các trường Header (GeneralResponseEntity) được theo sau bởi CRLF.

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

- Một phần thân thông báo tùy ý.

Các khu vực dưới đây giải thích cách sử dụng của mỗi đối tượng trong một thông báo phản hồi HTTP.

Dòng trạng thái

Một dòng trạng thái bao gồm phiên bản giao thức được theo sau bởi một mã hóa trạng thái số và cụm từ thuần văn bản được liên kết của nó.

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

Phiên bản HTTP

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

HTTP-Version = HTTP/1.1

Mã hóa trạng thái

Yếu tố Status-Code là một số nguyên 3 ký tự, trong đó ký tự đầu tiên của mã hóa trạng thái định nghĩa hạng (loại) phản hồi và hai ký tự cuối không có bất cứ vai trò phân loại nào. 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 được chấp nhận một cách thành công.
3	3xx: Sự điều hướng lại 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.
5	5xx: Lỗi Server Nó nghĩa là Server thất bại với việc thực hiện một yêu cầu nhìn như có vẻ khả thi.

Mã hóa trạng thái HTTP là có thể co giãn và ứng dụng HTTP không được yêu cầu để hiểu ý nghĩa của tất cả các mã trạng thái được đăng ký. Một danh sách của tất cả các mã trạng thái đã được cung cấp trong một chương riêng biệt cho bạn tham khảo.

Các trường Header Phản hồi

Chúng ta sẽ học General-Header và Entity-Header trong một chương riêng biệt khi chúng ta sẽ học về các trường Header. Bây giờ, chúng ta tìm hiểu xem các trường Header phản hồi là gì:

Các trường Header phản hồi cho phép Server truyền thông tin thêm về phản hồi mà không thể được đặt trong dòng Status-Line. Những trường Header này cung cấp thông 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

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

Các ví dụ về Thông báo Phản hồi

Bây giờ chúng ta đặt tất cả các thứ trên cùng với nhau để tạo một phản hồi HTTP cho một yêu cầu để chỉ thị trang hello.jsp từ Server đang chạy trên tutorialspoint.com.

```
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>
```

Ví dụ sau đây chỉ một thông báo phản hồi HTTP hiển thị trạng thái lỗi khi Server không thể tìm thấy trang được yêu cầu:

```
HTTP/1.1 404 Not Found Date: Sun, 18 Oct 2012 10:36:20 GMT Server:  
Apache/2.2.14 (Win32) Content-Length: 230 Connection: Closed Content-Type:  
text/html; charset=iso-8859-1
```

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN"> <html> <head> <title>404  
Not Found</title> </head> <body> <h1>Not Found</h1> <p>The requested URL  
/t.html was not found on this server.</p> </body> </html>
```

Tiếp theo là một ví dụ của một thông báo phản hồi HTTP chỉ trạng thái lỗi khi Server nhận vào một phiên bản HTTP sai trong yêu cầu HTTP đã cung cấp:

```
HTTP/1.1 400 Bad Request Date: Sun, 18 Oct 2012 10:36:20 GMT Server:  
Apache/2.2.14 (Win32) Content-Length: 230 Content-Type: text/html; charset=iso-  
8859-1 Connection: Closed  
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN"> <html> <head>  
<title>400 Bad Request</title> </head> <body> <h1>Bad Request</h1>  
<p>Your browser sent a request that this server could not understand.</p>  
<p>The request line contained invalid characters following the protocol  
string.</p> </body> </html>
```