



# Python: RESTful API application

ITFROVN - Python 2

# HTTP Protocol

02





## 2. HTTP - some

- URL vs URI.



- **MIME** type hay Media type (Multipurpose Internet Mail Extensions) do IANA chịu trách nhiệm ([link updated](#) [MIME type](#)): là một standard để xác định loại/định dạng của tài liệu (document).

.htm	HyperText Markup Language (HTML)	text/html
.html		

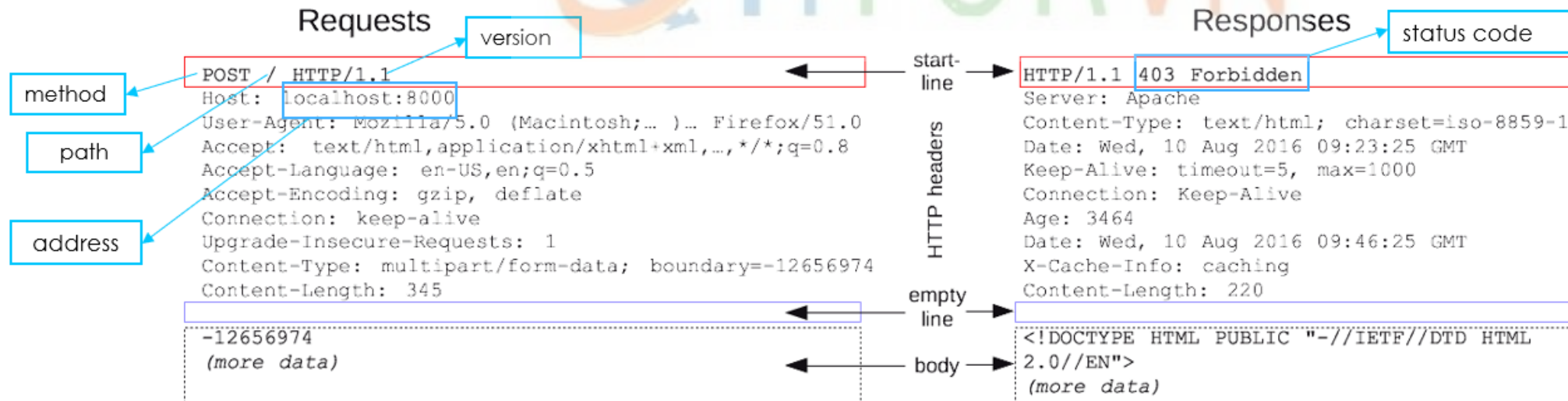


Browser sử dụng MIME type để xác định và quyết định cách xử lý với URL khi được load về. Browser không sử dụng file extension.

- **www vs non-www**: www không phải là một “bắt buộc” phải có với website. www về mặt kỹ thuật chỉ là một subdomain của domain name chính.

# 1. HTTP - Header

- HTTP client-protocol:
  1. Client khởi tạo TCP connection.
  2. Client gửi request, chờ đợi server gửi trả lại kết quả.
  3. Server process client request, phản hồi lại kết quả đã xử lý cho client: status code và dữ liệu đã xử lý.
- HTTP header updated list: <https://www.iana.org/assignments/message-headers/message-headers.xhtml>



# 1. HTTP – Cookies

- HTTP cookies (web cookies, browser cookies) – là các dữ liệu được gửi từ server xuống web browser và được web browser lưu lại, thường dùng cho các mục đích:
  - session management
  - tracking
  - personalization
- Cookie được set thông qua header “Set-Cookie” được gửi về từ server. Muốn set nhiều cookie thì có thể sử dụng nhiều header “Set-Cookie”. Một số lưu ý:
  - Expired time
  - Secure
  - HttpOnly
  - Domain and Path

**Set-Cookie:** uid=xuoeeekfk; Expires=Mon, 19 April 2021 07:28:00 GMT; Secure; HttpOnly

# 1. HTTP – Request methods

- Request methods – chỉ định các tác vụ mong muốn mà client muốn thực hiện trên một resource:
  - **GET**: thực hiện lấy resource mong muốn, và server chỉ trả về resource mà client cần.
  - **HEAD**: tương tự như GET nhưng không trả về resource data, chỉ có response header được trả về.
  - **POST**: submit data từ client đến server, thường dùng để tạo/thêm mới dữ liệu ở phía server.
  - **PUT**: thực hiện việc thay đổi hoàn toàn trên một đối tượng ở 1 resource cụ thể.
  - **PATCH**: thực hiện việc thay đổi 1 phần trên một đối tượng ở 1 resource cụ thể.
  - **DELETE**: xóa 1 resource cụ thể.
  - **CONNECT**
  - **OPTIONS**
  - **TRACE**

<b>C</b>	→	Create	→	<b>POST</b>
<b>R</b>	→	Read	→	<b>GET</b>
<b>U</b>	→	Update	→	<b>PUT</b>
<b>D</b>	→	Delete	→	<b>DELETE</b>

# 1. HTTP – Response status code

- HTTP response status code – giúp xác định một HTTP request được thực thi thành công hay thất bại, nếu thất bại thì lý do là gì.
- Status code được biểu thị bằng các số thập phân và được chia làm 5 nhóm:
  - Informational responses: 100-199
  - Successful responses: 200-299
  - Redirect: 300-399
  - Client errors: 400-499
  - Server errors: 500-599
- List status code: <https://datatracker.ietf.org/doc/html/rfc2616#section-10>



# 1. HTTP – Security

- HTTP Access control CORS (Cross-Origin Resource Sharing).

```
GET /resources/public-data/ HTTP/1.1
Host: bar.other
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:71.0) Gecko/20100101 Firefox/71.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Connection: keep-alive
Origin: https://foo.example
```

```
HTTP/1.1 200 OK
Date: Mon, 01 Dec 2008 00:23:53 GMT
Server: Apache/2
Access-Control-Allow-Origin: *
Keep-Alive: timeout=2, max=100
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: application/xml

[...XML Data...]
```

- SSL: Secure Socket Layer – Netscape thêm vào HTTP (năm 1994) “additional encrypted transmission layer”, gọi là SSL (phiên bản 1.0 internal company), SSL 2.0 và SSL 3.0.
- TLS: Transport Layer Security.
- CSP: Content-Security-Policy – cho phép web administrator control client được/không được load một trang tài liệu nào đó.

```
Content-Security-Policy: frame-ancestors none;
Content-Security-Policy: frame-ancestors <source> <source>;
```