

JEE Course HTTP Protocol

Márcio Fuckner
marcio.fuckner@hva.nl

HvA – Hogeschool van Amsterdam

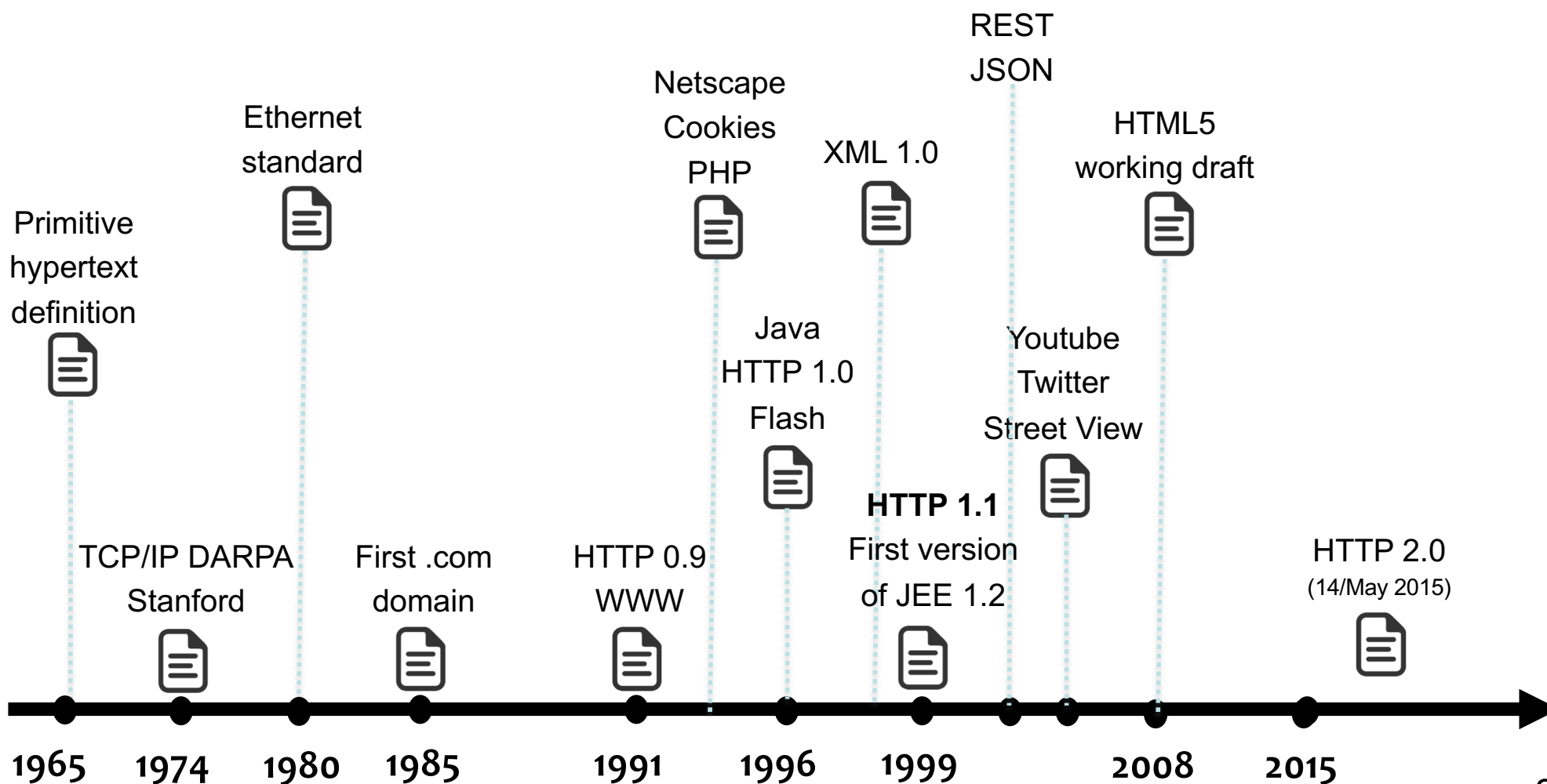


Content

- Brief history
- Main concepts
- Request and response messages
- Status codes
- Headers
- MIME



Timeline of HTTP and related technologies





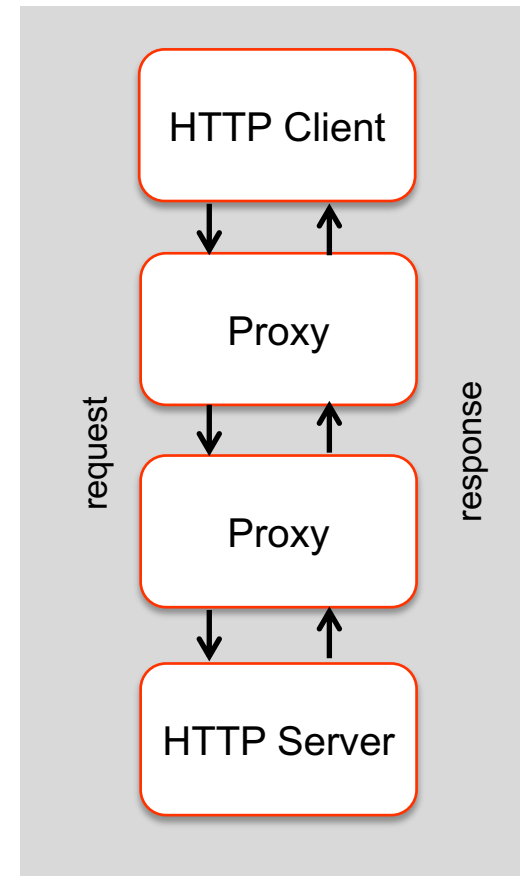
HTTP features (1 of 2)

- ▶ According to Berners-Lee, the world wide web has three key elements
 - ▶ A **markup language** to represent the information
 - ▶ A **common notation** to access resources
 - ▶ A **transport protocol**
- ▶ Markup languages: The most popular are **HTML**, XML and their variations.
- ▶ The conventional notation is called **URI** (*Uniform Resource Identifier*), and can be found using a URL (*Uniform Resource Locator*).
- ▶ **HTTP** is the transport protocol



HTTP features (2 of 2)

- HTTP is a **TCP/IP based** communication protocol used to deliver data on the web
- HTTP 2.0 is described in the **RFC 7540**
- Request and response paradigm
- Stateless protocol



Uniform Resource Locator (1 of 3)



- ▶ Uniform Resource Locators (URLs) are strings used to find a resource.
- ▶ General notation:

protocol://**host**[:**port**]/**path**[:**parameters**][?**query**][#**anchor**]

- ▶ **protocol**: Name of the protocol (Ex.: HTTP, HTTPS, e FTP). The name precedes the sequence “://”;
- ▶ **host**: IP address or hostname;
- ▶ **port**: Optional. Indicates what port is used to connect the server. If the port is omitted, then the standard port is used. For example, the default port for the HTTP protocol is 80;
- ▶ **path**: absolute path of the resource. In general, some HTTP servers allow the creation of aliases, which does not mean that the resource is navigable in the server file system.



Uniform Resource Locator (2 of 3)

- ▶ Uniform Resource Locators (URLs) are strings used to find a resource.
- ▶ General notation:

protocol://**host**[:**port**]/**path**[:**parameters**][?**query**][#**anchor**]

- ▶ **parameters**: Optional. Commonly used for session tracking.
- ▶ **query**: Optional. A list of parameters (key=value pairs), separated by the ampersand (&) character
- ▶ **anchor**: Optional. Determines a specific part of the document (like a bookmark)

Uniform Resource Locator (3 of 3)



► Some examples of URL:

<http://www.google.com.br/search?hl=en&q=Web+Technology&aq=f&oq=>

- Protocol: HTTP
- Host: www.google.com.br
- Port: Omitted (80)
- Path: /search
- Query: hl=en&q=Web+Technology&aq=f&oq=

<ftp://ftp.pucpr.br:21/FreeBSD>

- Protocol: ftp
- Host: ftp.pucpr.br
- Port: 21
- Path: /FreeBSD



HTTP request

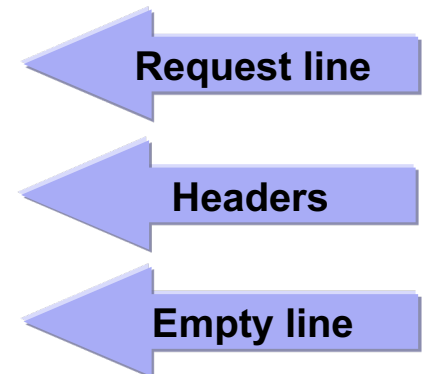
- When we open the browser and type a URL, the browser connects to the specified host and sends an HTTP request message
- An HTTP request have the following format:

Method path+parameters+query+anchor HTTP/version

Name-of-the-header-1: value-1

Name-of-the-header-n: value-n

[optional body]





HTTP methods

- ▶ There are several HTTP request methods available
 - ▶ *GET*: used to retrieve information
 - ▶ *POST*: posting a message, providing blocks of data
 - ▶ *PUT*: Add a resource
 - ▶ *DELETE*: Remove a resource
 - ▶ *TRACE*: Use for testing and diagnose purposes
 - ▶ *OPTIONS*: request for information
 - ▶ *CONNECT*: For use with a proxy

Hint: During the beginning of the 2000's, GET and POST were the most used HTTP methods. With the adoption of lightweight web services following the REST specification, other methods such as PUT and DELETE have gained more attention



GET method example

- ▶ When we click on a link or when we type a URL in the browser, it creates an HTTP request message using GET
- ▶ GET sends request data using the "?" character to mark the beginning of the requests and "&" to delimit multiple parameters
- ▶ Example:
- ▶ When typing the following URL into your browser:
 - ▶ `http://www.google.com` and enter the search value "Metropolia", the browser will send the following request line:
- ▶ `GET /?gws_rd=ssl#q=Metropolia&safe=off HTTP/1.1`



POST method example

- ▶ HTML forms submitted using the post method produce an HTTP message the query part of the body of the HTTP message.
- ▶ Remember that all HTTP messages are allowed to have a body!
- ▶ Example:

POST /q HTTP/1.1 ← **request line**

Host: fun.yahoo.com

User-Agent: Mozilla/5.0 (Macintosh; ...

Content-Type: application/x-www-form-urlencoded

Content-Length: 27

search=restaurants&count=10 ← **body**

← **headers**



HTTP response

► General format

HTTP/version status code message

Header-1: valor-1

Header-2: valor-n

[optional body]



Status Codes

- ▶ Range of status codes
 - ▶ 1xx Information (used for proxy chaining)
 - ▶ 2xx Success
 - ▶ 3xx Redirection
 - ▶ 4xx Client error
 - ▶ 5xx Server error



Headers (1 of 2)

- Headers are used to send additional information (to the server or the client)
- Used to improve the HTTP communication (connection optimization, cache, content-types, session tracking)
- A set of predefined headers is available. But you are free to create your own set of headers



Headers (2 of 2)

General headers

Date: Sun, 01 Mar 2015 11:05:48 GMT

Connection: Close

Request headers

User-Agent: Mozilla/5.0 ...

Authorization: Basic [cred.]

Accept-Language: en-us,en;q=0.5

Response headers

Location: www.haschanged.com

www-Authenticate: Basic realm="XFiles"

Server: Apache/2.4.12

Entity Headers

Last-Modified: Sun, 01 Mar 2015 ...

Content-Type: application/pdf

Content-Length: 250



Content-Type Header

- ▶ How to parse the body content? What is the format?
- ▶ HTTP has used the ideas applied to e-mail messages.
- ▶ An email message may have a body in pure textual format or HTML and several attachments
- ▶ The MIME standard (*Multipurpose Internet Mail Extension*) used for e-mail purposes is also used for HTTP messages
- ▶ Format: type/subtype

Examples: text/html, text/plain, application/json, text/xml

See: IANA media types: at <http://www.iana.org/assignments/media-types/media-types.xhtml>



Further reading

- ▶ RFC 7540 (*Hypertext Transfer Protocol – HTTP/2*)
<http://www.ietf.org/rfc/rfc7540.txt>
- ▶ RFC 1738 (*Uniform Resource Locators*)
<http://www.w3.org/Addressing/rfc1738.txt>
- ▶ IANA media types
<http://www.iana.org/assignments/media-types/media-types.xhtml>
- ▶ RFC 2045 (*Multipurpose Internet Mail Extensions*)
<http://www.ietf.org/rfc/rfc2045.txt>