COMP 1842

Week 2 – Part 1- HTTP

Matt Prichard

Introduction

Clients and Servers

HTTP

View headers - plug in

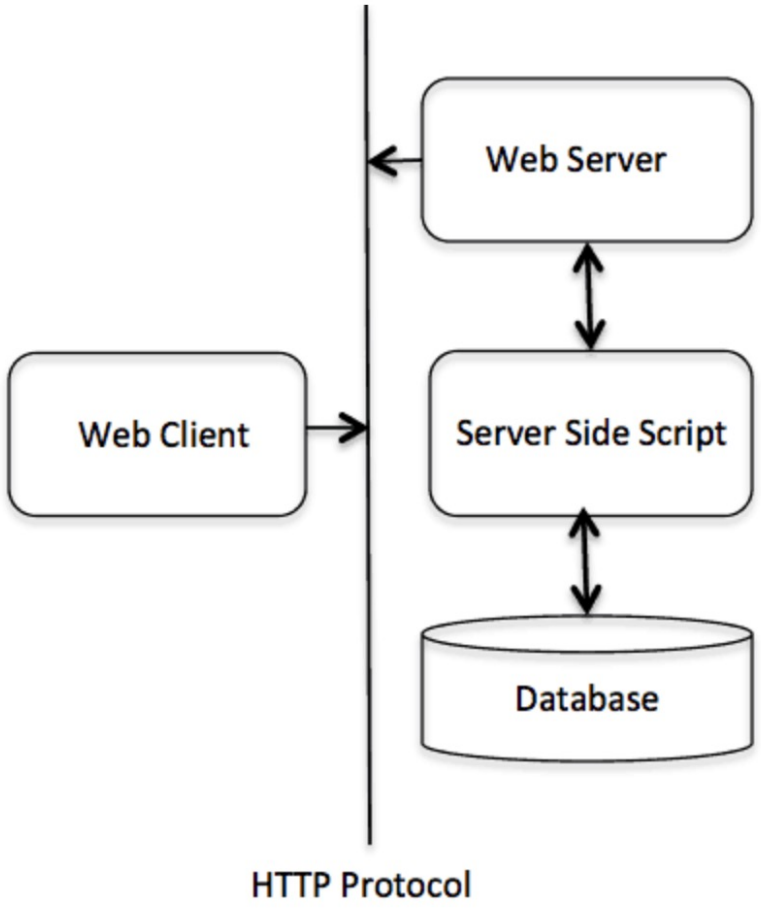
Error codes

HTTP

Hypertext Transfer Protocol

Basic Architecture

The following diagram shows a very basic architecture of a web application and depicts where HTTP sits:



The HTTP protocol is a request/response protocol based on the client/server based architecture where web browsers, robots and search engines, etc. act like HTTP clients, and the Web server acts as a server.

Client: The HTTP client sends a request to the server in the form of a request method, URI, and protocol version, followed by a MIME-like message containing request modifiers, client information, and possible body content over a TCP/IP connection.

Server: The HTTP server responds with a status line, including the message's protocol version and a success or error code, followed by a MIME-like message containing server information, entity meta information, and possible entity-body content.

Client: the user-agent

- The user-agent is any tool that acts on behalf of the user. This role is primarily performed by the Web browser.
- To display a Web page, the browser sends an original request to fetch the HTML document that represents the page. It then makes additional requests corresponding to execution scripts, layout information (CSS) to display, and sub-resources contained within the page (usually images and videos).
- The Web browser then combines these resources to present the complete document, the Web page. Scripts executed by the browser can fetch more resources in later phases and the browser updates the Web page accordingly.

The web server

- On the opposite side of the communication channel is the server, which serves the document as requested by the client.
- A server appears as only a single machine virtually; but it may actually be a collection of servers sharing the load (load balancing), or a complex piece of software interrogating other computers (like cache, a DB server, or e-commerce servers), totally or partially generating the document on demand.
- A server is not necessarily a single machine, but several server software instances can be hosted on the same machine.

Proxies

Between the Web browser and the server, numerous computers and machines relay the HTTP messages. Those operating at the application layers are generally called **proxies**. These can be transparent, forwarding on the requests they receive without altering them in any way, or non-transparent, in which case they will change the request in some way before passing it along to the server. **Proxies** may perform numerous functions:

- caching (the cache can be public or private, like the browser cache)
- filtering (like an antivirus scan or parental controls)
- authentication (to control access to different resources)
- logging (allowing the storage of historical information)

HTTP Basic features

HTTP is connectionless: The HTTP client, i.e., a browser initiates an HTTP request and after a request is made, the client waits for the response. The server processes the request and sends a response back after which the client disconnects the connection. So the client and server knows about each other during current request and response only. Further requests are made on new connections as if client and server are new to each other.

HTTP Basic features 2

HTTP is media independent: It means, any type of data can be sent by HTTP as long as both the client and the server know how to handle the data content. It is required for the client as well as the server to specify the content type using appropriate MIME-type (see next slide).

HTTP is stateless: As mentioned above, HTTP is connectionless and as a direct result of HTTP being a stateless protocol. The server and client are aware of each other only during a current request. Afterwards, both of them forget about each other. Due to this nature of the protocol, neither the client nor the browser can retain information between different requests across the web pages.

MIME type

A media type (also known as a **Multipurpose Internet Mail Extensions** or **MIME type**) indicates the nature and format of a document, file, or assortment of bytes.

https://www.sitepoint.com/mime-types-complete-list/

Important MIME types for Web developers include:

text/plain

text/html

text/css

text/javascript

Image/jpeg

image/png

Audio/aiff…etc etc

https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/MIME_types

http flow

When a client wants to communicate with a server it performs the following steps:

1. Open a TCP connection: The TCP connection is used to send a request, or several, and receive an answer.
2. Send an HTTP message: HTTP messages (before HTTP/2) are human-readable. With HTTP/2, these simple messages are encapsulated in frames, making them impossible to read directly, but the principle remains the same. For example:

```
GET / HTTP/1.1
Host: developer.mozilla.org
Accept-Language: fr
```

http flow...cont

3. Read the response sent by the server, such as:

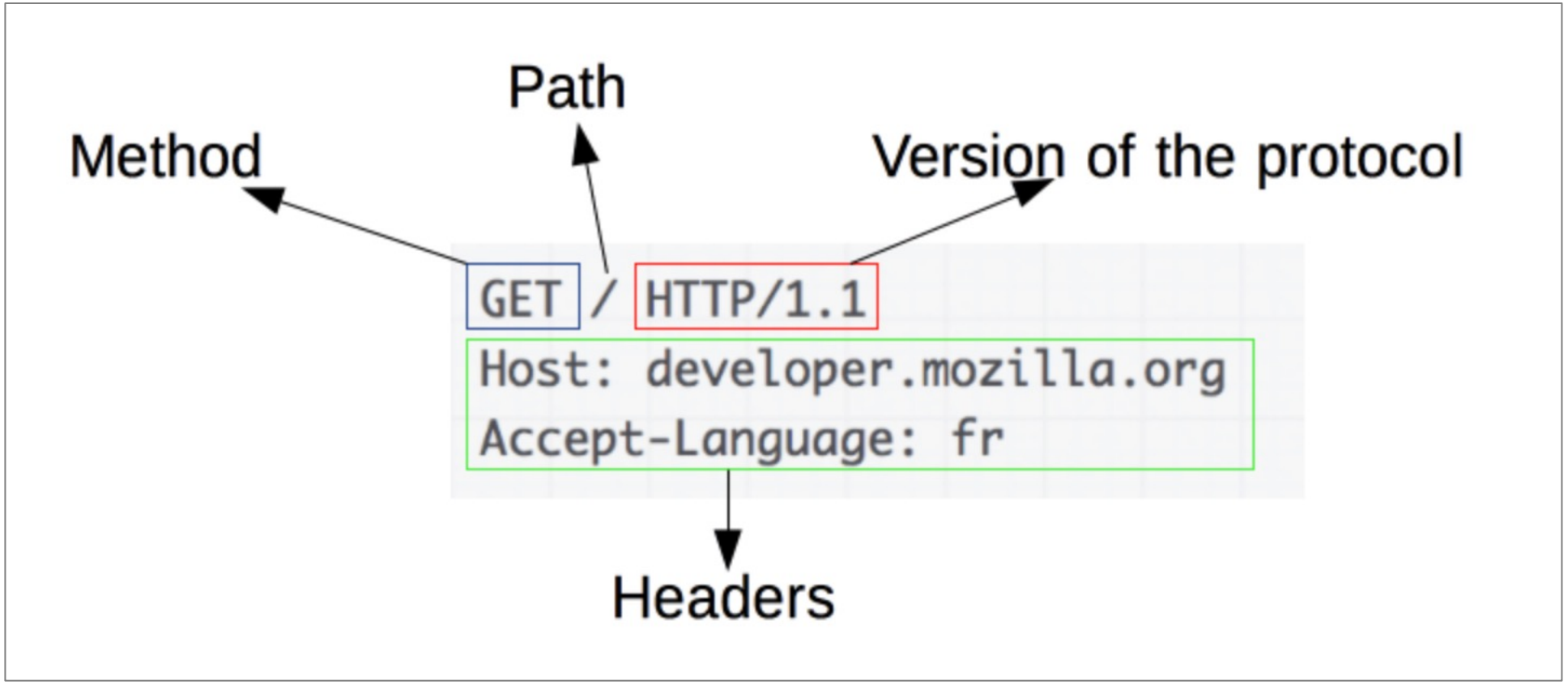
```
HTTP/1.1 200 OK
Date: Sat, 09 Oct 2010 14:28:02 GMT
Server: Apache
Last-Modified: Tue, 01 Dec 2009 20:18:22 GMT
ETag: "51142bc1-7449-479b075b2891b"
Accept-Ranges: bytes
Content-Length: 29769
Content-Type: text/html

<!DOCTYPE html... (here come the 29769 bytes of the requested web page)
```

4. Close or reuse the connection for further requests.

Requests

There are two types of HTTP messages, requests and responses, each with its own format.

An example HTTP request:

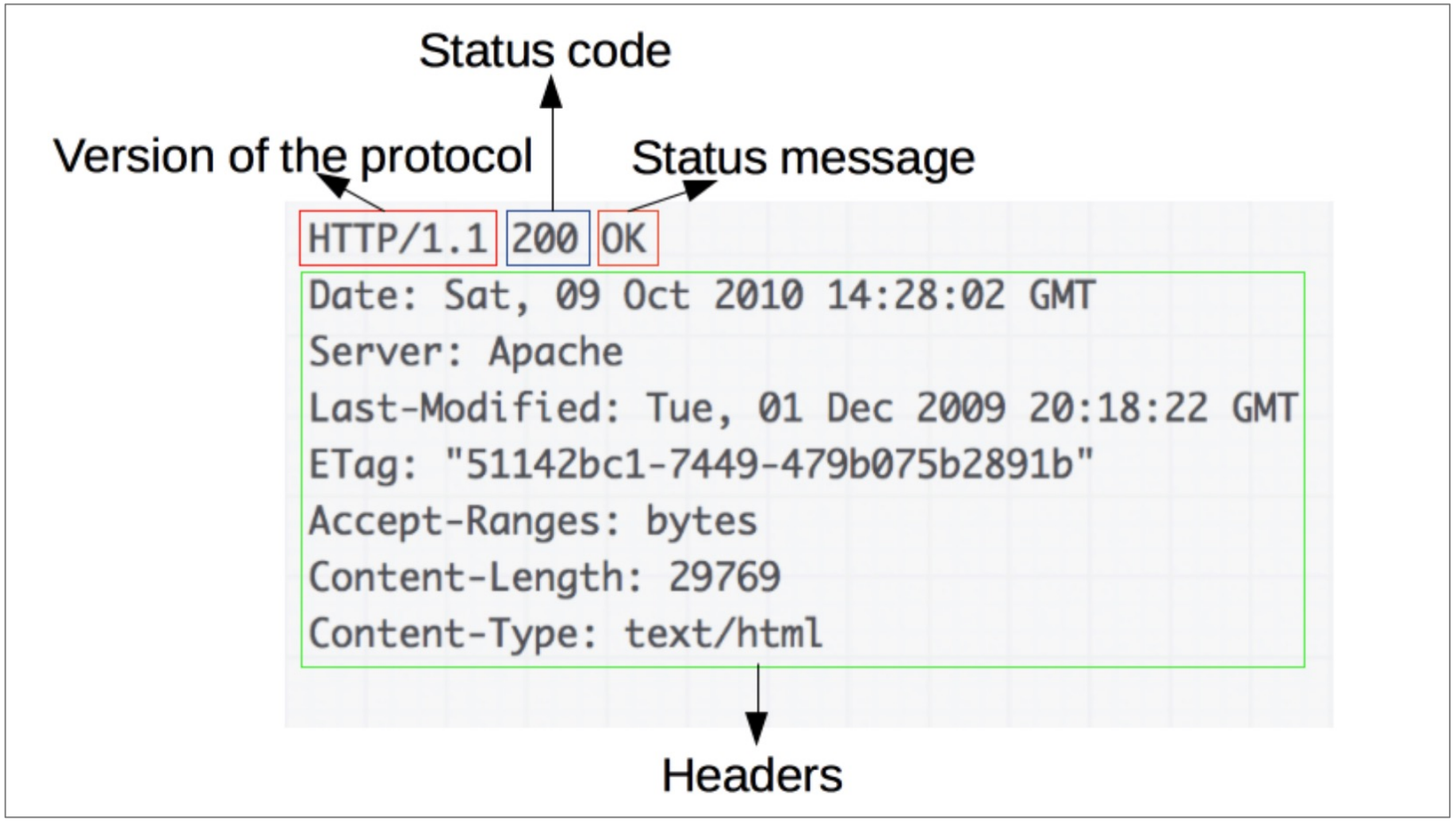


Requests...cont

- Requests consists of the following elements:
- An HTTP method, usually a verb like GET, POST_that defines the operation the client wants to perform. Typically, a client wants to fetch a resource (using GET) or post the value of an HTML form (using POST), though more operations may be needed in other cases.
- The path of the resource to fetch.
- The version of the HTTP protocol.
- Optional headers that convey additional information for the servers.
- A body, for some methods like POST, similar to those in responses, which contain the resource sent.

Responses

An example HTTP response:



Responses...cont

- Responses consist of the following elements:
- The version of the HTTP protocol they follow.
- A **status code**, indicating if the request was successful or not, and why.
- A status message, a non-authoritative short description of the status code.
- HTTP **headers**, like those for requests.
- Optionally, a body containing the fetched resource.

HTTP request headers

Some of the most common HTTP headers found in HTTP requests.

```
Host: code.tutsplus.com
```

This is basically the host name, including the domain and the subdomain.

```
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1;
```

Browser name and version, operating system name and version etc

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

This header displays the default language setting of the user.

```
Accept-Encoding: gzip,deflate
If-Modified-Since: Sat, 28 Nov 2009 06:38:19 GMT
Cookie: PHPSESSID=r2t5uvjq435r4q7ib3vtdjq120; foo=bar
Authorization: Basic bXl1c2VyOm15cGFzcw==
```

https://code.tutsplus.com/tutorials/http-headers-for-dummies--net-8039

HTTP response headers

Content-Type- This header indicates the "MIME type" of the document. The browser then decides how to interpret the contents based on this. For example, an HTML page (or a PHP script with HTML output) may return this:

```
Content-Type: text/html; charset=UTF-8
```

For a GIF image, this may be sent:

```
Content-Type: image/gif
```

When content is going to be transmitted to the browser, the server can indicate its size (in bytes) using this header.

```
Content-Length: 89123
```

As the name suggests, this header indicates the last modify date of the document, in GMT format:

```
Last-Modified: Sat, 28 Nov 2009 03:50:37 GMT
```

There are many more - see:

https://www.tutorialspoint.com/http/http_header_fields.htm

HTTP status codes

S.N.	Code and Description
1	**1xx: Informational** It means the request has been received and the process is continuing.
2	**2xx: Success** It means the action was successfully received, understood, and accepted.
3	**3xx: Redirection** It means further action must be taken in order to complete the request.
4	**4xx: Client Error** It means the request contains incorrect syntax or cannot be fulfilled.
5	**5xx: Server Error** It means the server failed to fulfill an apparently valid request.

The Status-Code element in a server response, is a 3-digit integer where the first digit of the Status-Code defines the class of response. There are 5 values for the first digit.

Common status codes – 2s

200 OK

As mentioned before, this status code is sent in response to a successful request.

206 Partial Content

If an application requests only a range of the requested file, the 206 code is returned. It's most commonly used with download managers that can stop and resume a download, or split the download into pieces.

Common status codes – 3s

302 (or 307) Moved Temporarily & **301** Moved Permanently

These two codes are used for redirecting a browser. For example, when you use a URL shortening service, such as bit.ly, that's exactly how they forward the people who click on their links.

Both 302 and 301 are handled very similarly by the browser, but they can have different meanings to search engine spiders. For instance, if your website is down for maintenance, you may redirect to another location using 302. The search engine spider will continue checking your page later in the future

Common status codes – 4s

Everyone's favourite

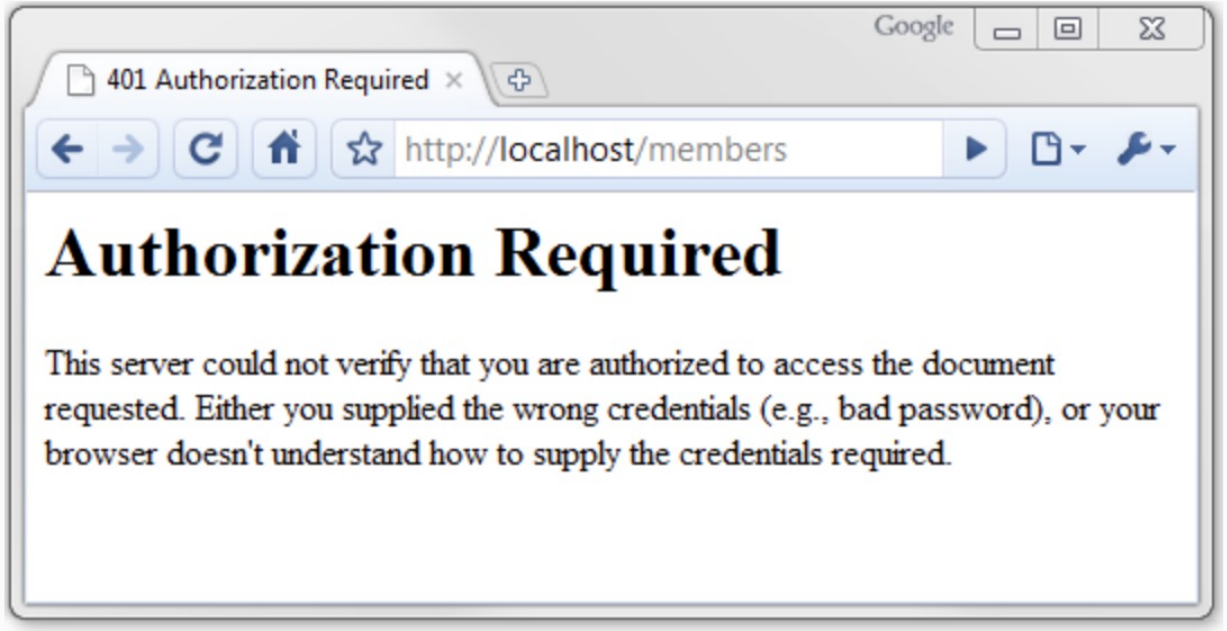
404 Not Found

When the requested page or file was not found, a 404 response code is sent by the server.

Common status codes – 4s

401 Unauthorized

Password-protected web pages send this code. If you don't enter a login correctly, you may see the following in your browser.



Common status codes – 4s

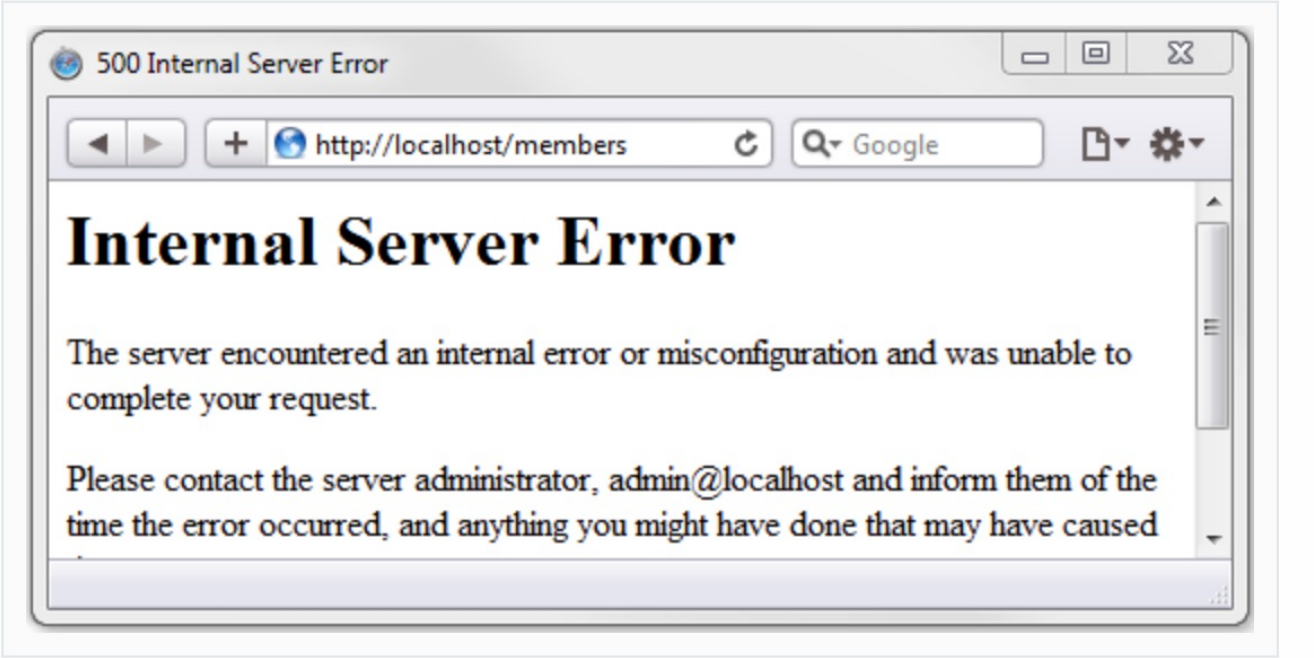
403 Forbidden

If you are not allowed to access a page, this code may be sent to your browser. This often happens when you try to open a URL for a folder that contains no index page. If the server settings do not allow the display of the folder contents, you will get a 403 error.

Common status codes – 5s

Very common when developing and unhelpful ☹

500 Internal Server Error



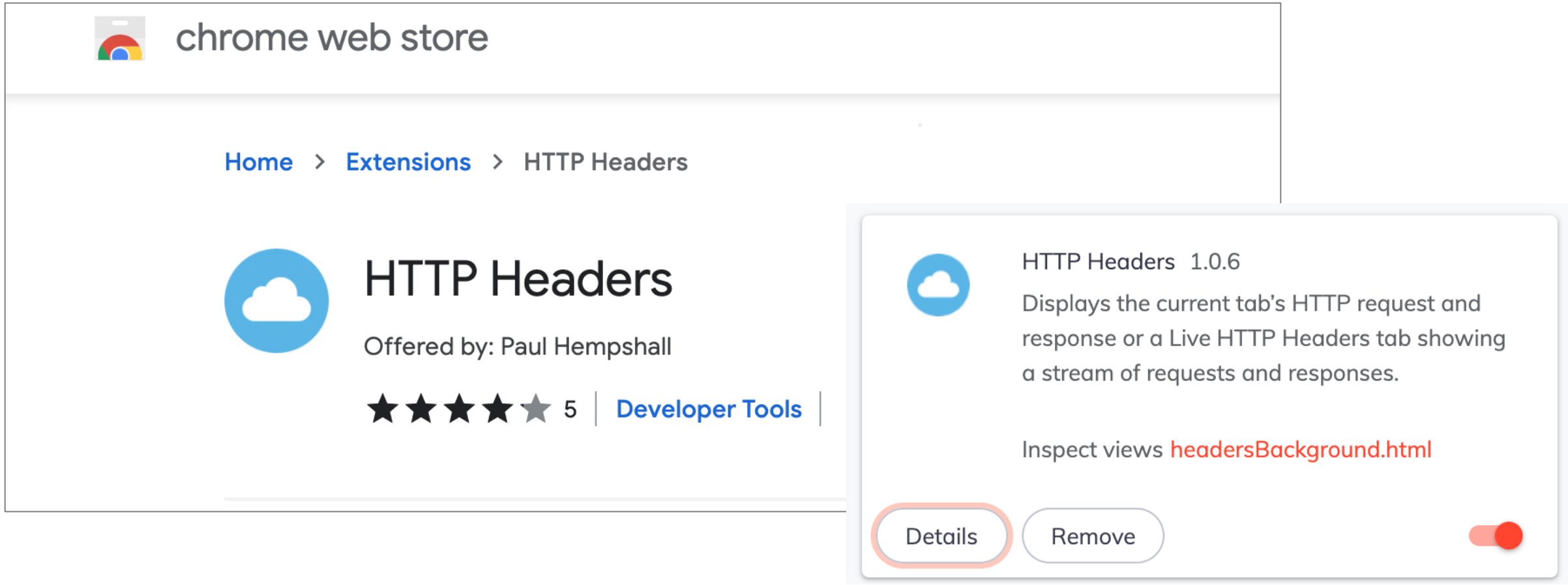
This code is usually seen when a web script crashes. Most CGI scripts do not output errors directly to the browser, unlike PHP. If there are any fatal errors, they will just send a 500 status code. And the programmer then needs to search the server error logs to find the error messages.

Complete List

You can find the complete list of HTTP status codes with their explanation on Wikipedia.

HTTP headers browser extension

see video for demo



References/sources

Developer.mozilla.org. 2022. HTTP | MDN. [online] Available at: <https://developer.mozilla.org/en-US/docs/Web/HTTP/> [Accessed 25 January 2022].

Guzel, B., 2022. HTTP Headers for Dummies. [online] Code Envato Tuts+. Available at: <https://code.tutsplus.com/tutorials/http-headers-for-dummies--net-8039> [Accessed 25 January 2022].

Tutorialspoint.com. 2022. HTTP - Responses. [online] Available at: <https://www.tutorialspoint.com/http/http_responses.htm> [Accessed 25 January 2022].