⌂ / [Dashboard](#) / My courses / [Programming in Python - 2. Python Programming](#) / [4. Programming for Computer Networks](#)

/ [4.2 Introduction to HTTP](#)

4.2 Introduction to HTTP

⌂ / [Dashboard](#) / My courses / [Programming in Python - 2. Python Programming](#) / [4. Programming for Computer Networks](#)

/ [4.2 Introduction to HTTP](#)

📄 **Summary & submit**

Introduction to HTTP

HTTP- Hyper Text Transfer Protocol is the application level protocol to communicate between web servers and web browsers. It provides the means for web browser to request content from a web server using the **GET** method, and sending content to a web server using **POST** method. Other methods such as OPTIONS, PUT and DELETE are also specified in the HTTP protocol.

GET method

Get request must be a string, which is framed as the following example:

GET / HTTP/1.1
Host:example.com
Connection:close

At the end of the string, there must be two new line (\n) characters. There are three segments in first line. Between each specific segment in the first line, there must be only one 'space'

Here the specifics are as follows:

- GET is the method

- / is the resource which is being requested from the server. / usually points to the index.html (or index.php etc.) file. If you need a page by any other name, it can be specified here.

- HTTP/1.1 is the protocol version. Many modern web sites uses HTTP/2 as well.

- Next lines (apart from the first line) are HTTP headers. There are many optional headers which can be used here, for example User-Agent, Cookie and Accept

GET method parameters

The GET method's requested resource can be subjected to parameters.

Example: You can request google.com/search resource for the search query 'Sri Lanka' and language preference of Sinhala as follows:

https://www.google.com/search?q=Sri+Lanka&hl=SI

Here the GET request will look something like this.

GET /search?q=Sri+Lanka&hl=SI HTTP/2
Host: www.google.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:98.0) Gecko/20100101 Firefox/98.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

Now we are passing two parameters in this request. They are q and hl. Parameters must be appended to the resource name, after a ? character. Parameters are arranged as name=value pair. q=Sri+Lanka and hl=SI. Now, since we have multiple parameters, those can be concatenated with the & character. Therefore, the full resource query parameters are q=Sri+Lanka&hl=SI
Full resource query is /search?q=Sri+Lanka&hl=SI

Exercise 1

Now, check the search results of the following two links in the web browser, and identify the differences.

https://www.google.com/search?q=Python&hl=SI
https://www.google.com/search?q=Python&hl=EN

Once you are at the web page, Press Ctrl+Shift+I in the browser, Select Network Tab to open the Network Monitoring Tool. You can reload the page by pressing enter after placing the cursor at the address bar. When the page is re-loading you can observer the Network Moitoring Tool listing the HTTP

transactions one by one.

Scroll all the up in the in the network moitoring tool, and select the first GET request. Then, locate the request headers (and enable raw format) to study the full request.

Follow the numbers in the following image to locate and study the actual GET method request body.
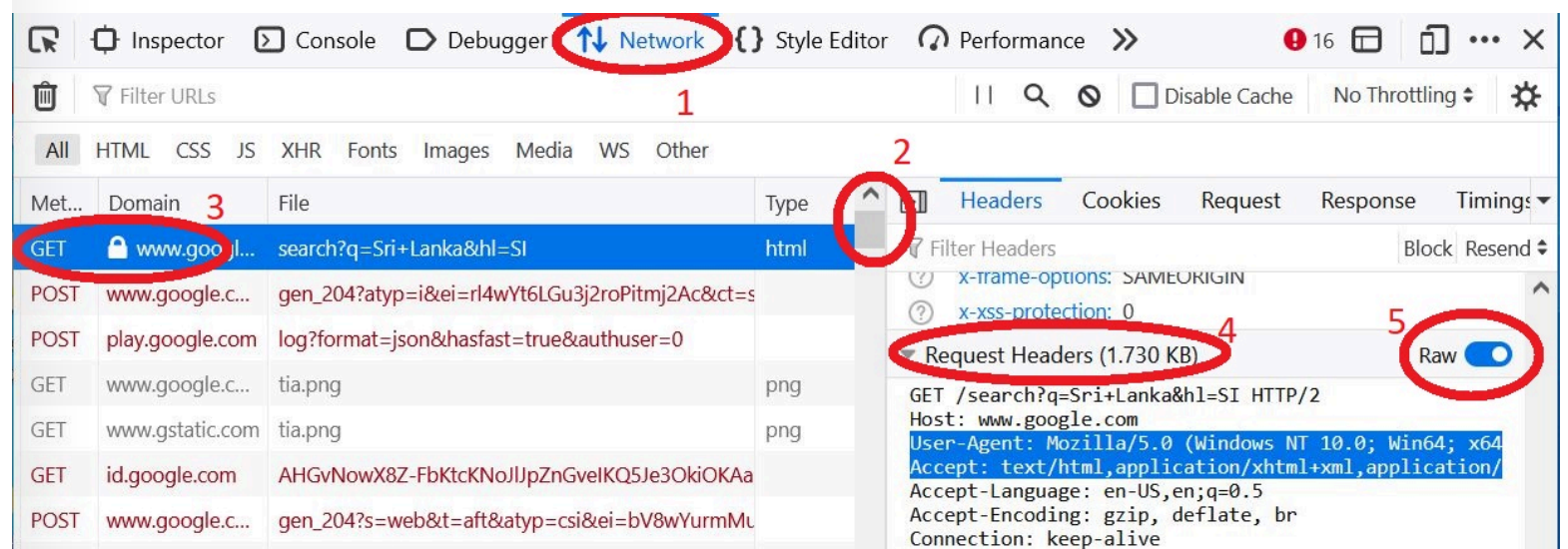
HTTP Status Codes

Various Response Status Codes could appear in the HTTP responses.

200 OK is the most common response, which means the request has been fulfilled.

You will frequently come across some of the following as well

- 301 Moved Permanently

- 302 Found/Moved Temporarily

- 403 Forbidden

- 404 Not Found

- 451 Unavailable

If you find some unfamiliar status code, you can find the details about all types of status codes in a google search, or in Wikipedia.

https://en.wikipedia.org/wiki/List_of_HTTP_status_codes

Exercise 2

One way to test the HTTP protocol better is to do it on top of a normal TCP socket. Let us try to connect to the web server example.com and retrieve a web page using the HTTP protocol

POST method

POST method is used to send content to the server.

POST / HTTP/1.1
Host:example.com
Content-Length:0
Connection:close

"Payload file"

The content like data such as username, password, or even images can be appended to the POST request at the location of "payload file"

Exercise 3

Try a POST method using a TCP socket to the exampe.com:80

≡ | 🐍 3 | ✏️ _Code | ▶ Run | ▼ | ❮ Share | ▼ | 💾

‹ › **matrinket** 🔑 Python3 Remix ✛ ⬆

Choose File | No file chosen

Jump to...

GET IN TOUCH

🏠 University of Moratuwa
Centre for Open & Distance Learning
CODL

📞 011 308 2787/8

📞 011 265 0301 ext. 3850,3851

✉ open@uom.lk

- 🌐 [University Website](#)
- 🌐 [CODL Website](#)

[Data retention summary](#)