## **Part One: Solidify Terminology**

In your own terms, define the following terms:

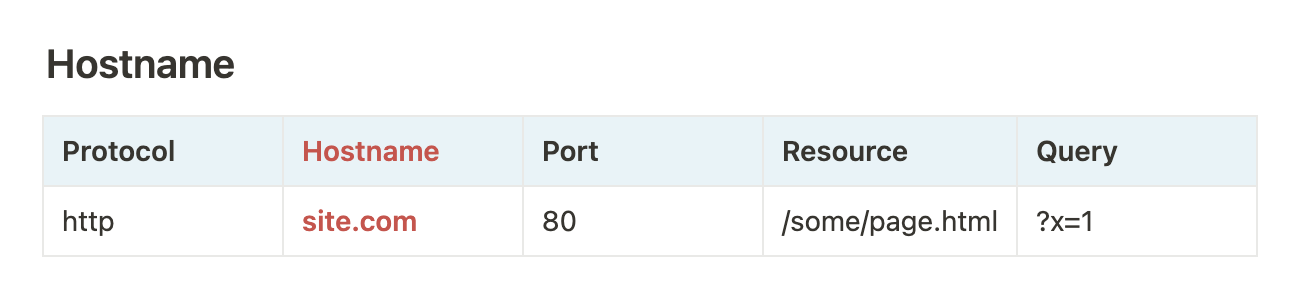
* **What is HTTP?**

It stands for Hypertext Transfer Protocol and it’s a way of requesting information over a network by communication platforms such as web browsers when they need to load a website from a server.

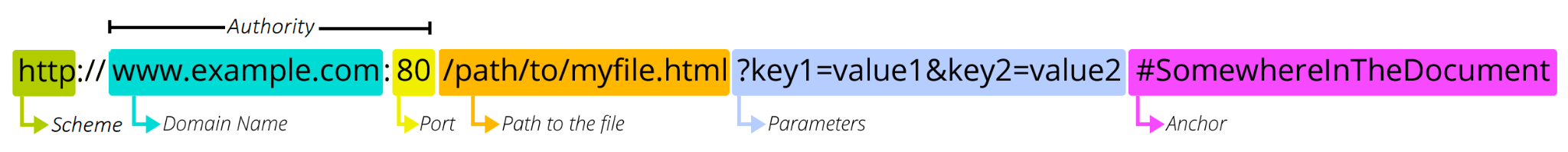
HTTPS is HTTP with encryption and verification. The only difference between the two protocols is that HTTPS uses TLS (SSL) to encrypt normal HTTP requests and responses, and to digitally sign those requests and responses. That’s why HTTPS is far more secure than HTTP

* **What is a URL?**

URL stands for Uniform Resource Locator. It’s an address given to a unique resource on the web.



URL Anatomy of an absolute URL:



<https://developer.mozilla.org/en-US/docs/Learn/Common_questions/Web_mechanics/What_is_a_URL>

^^ for more information about the anatomy of an url.

* **What is DNS?**

DNS stands for Domain Name System. It’s like a phonebook of the Internet. It translates domain names to IP addresses so browsers can load Internet resources. Every device connected has a unique IP address that other machines use to find the device. DNS servers eliminate the need for humans to memorize complex IP addresses to use the web.

* **What is a query string?**

Query strings are part of an url that provides “extra information” – search terms, info from forms, etc. The server is provided this information and it uses this information to change the page.

Multiple arguments are separated by &: ?x=1&y=2

* Arguments can be given several times: ?x=1&x=2

The query string allows you to pass key-value pairs into the URL, in the format ?key1=value1&key2=value2...

* **What are two HTTP verbs (aka methods) and how are they different?**

HTTP requests has two main request methods that it uses to fetch data.

**GET**: requests without side effects (ie, don’t change server data)

* + Typically, arguments are passed along in query string
    - If you know the arguments, you can change the URL
  + Entering-URL-in-browser, clicking links, and *some* form submissions

**POST:** requests with side effects (ie change data on server)

* + Typically, arguments sent as body of the request (not in query string)
  + *Some* form submissions (but never entering-URL-in-browser or links)
  + Always do this if there’s a side-effect: sending mail, charge credit card, posts on websites etc
  + “Are you sure you want to resubmit?”
* **What is an HTTP request?**

HTTP request is when an individual web browser or device is asking for information from a server using the GET method. The request contains several key information when making the request.

### What’s in a Request?

* Method (ex: ***GET***)
* HTTP protocol version (almost always 1.1)
* Resource URL you want
* Headers
  + Hostname you’re asking about
  + Date your browser thinks it is
  + Language your browser wants information in
  + Any cookies that server has sent
  + And more!
* **What is an HTTP response?**

HTTP response is the reponse that the server sends back in response to a GET request. The server itself would have to interact with other things such as database servers and send back the information you want.

### What’s in a Response

* HTTP protocol version (almost always 1.1)
* Response Status Code (200, 404, etc)
* Headers
  + Content Type (typically text/html for web pages)
  + Date/time the server thinks it is
  + Any cookies server wants to set
  + Any caching information
  + And more!
* **What is an HTTP header? Give a couple examples of request and response headers you have seen.**

Headers provide additional information about the request or the response. Here are some examples:

* Request headers: Host, User-Agent, Accept, Cookie, Cache-Control
* Response headers: Content-Type, Last-Modified, Set-Cookie, Cache-Control
* **What are the processes that happen when you type “**[**http://somesite.com/some/page.html”**](http://somesite.com/some/page.html%E2%80%9D) **into a browser?**

Your browser “resolves” the name into an IP address using DNS

1. Your browser makes a request to that IP address, including headers (info about browser, any previous cookies, and other things)
2. The server sends a response (typically, HTML, with a status code (200 if it was sucessful)
3. The browser makes a DOM from that HTML, and finds any other resources needed (images, CSS, JavaScript, etc)
4. The browser makes separate HTTP requests for those resources and receives response from the server for each