

18.1 How Web Works Exercise

Part One: Solidify Terminology

In your own terms, define the following terms:

- What is HTTP?
 - HTTP stands for Hyper Text Transfer Protocol, which dictates how data is sent to or is received to a server.
- What is a URL?
 - URL stands for Uniform Resource Locator, it is an address that is used to locate an internet resource.
- What is DNS?
 - DNS stands for Domain Name System, “the phonebook of the internet”, it converts URLs into IP addresses.
- What is a query string?
 - A query string is a set of letters at the end of a URL, it allows you to filter and organize content.
- What are two HTTP verbs and how are they different?
 - Two HTTP terms are GET and POST. GET gets data while POST sends data.
- What is an HTTP request?
 - A request made by a user to the server.
- What is an HTTP response?
 - The response from the server to the user's request.
- What is an HTTP header? Give a couple examples of request and response headers you have seen.
 - Headers allow for additional information within the request or response.
 - Request examples: Accept, Cookie, Connection.
 - Response examples: Last-Modified, Expires, Content-Type.
- What are the processes that happen when you type [“http://somesite.com/some/page.html”](http://somesite.com/some/page.html) into a browser?
 - Browser uses DNS to turn the name into an IP address.
 - Browser makes a request to the IP address.
 - The server then sends a response.
 - Browser will then make a DOM from the HTML.
 - Lastly, browser will make additional requests for resources and will receive responses from each of them.

Part Two: Practice Tools

1. Using **curl**, make a **GET** request to the icanhazdadjoke.com API to find all jokes involving the word "pirate"
 - a. `curl -H "Accept: application/json" "https://icanhazdadjoke.com/search?term=pirate"`
2. Use **dig** to find what the IP address is for icanhazdadjoke.com.
 - a. `dig icanhazdadjoke.com`. IP address = 172.67.198.173
3. Make a simple web page and serve it using **python3 -m http.server**. Visit the page in a browser.

Part Three: Explore Dev Tools (completed)

Build a very simple HTML form that uses the GET method (it can use the same page URL for the action) when the form is submitted.

Add a field or two to the form and, after submitting it, explore in Chrome Developer tools how you can view the request and response headers.

Edit the page to change the form type to POST, refresh in the browser and re-submit. Do you still see the field in the query string? Explore in Chrome how you can view the request and response headers, as well as the form data.

Part Four: Explore the URL API (completed)

At times, it's useful for your JavaScript to look at the URL of the browser window and change how the script works depending on parts of that (particularly the query string).

[Read about the URL API](#)

Try some of the code examples in the Chrome Console so that you can get comfortable with the basic methods and properties for instances of the URL class.