HTTP

HTTP is the protocol that web browsers use to talk to web servers. This page will give you a brief overview.

URL

The web is designed to help web browsers load resources from web servers. These resources are named with Uniform Resource Identifiers, or URIs. Originally, URIs were designed quite generally, but today we usually use only one type of URI, called a Uniform Resource Locator, or URL.

For example, a URL could be:

```
https://www.amazon.com/CSS-Missing-David-Sawyer-McFarland/dp/1491918055
```

This URL starts with a scheme, in this case HTTPS, which means the secure form of the HTTP protocol. This will direct the browser to connect to the website on port 443 and negotiate a secure connection.

Next the URL has host, www.amazon.com, which is the name of the website to connect to.

Finally, there is a path, CSS-Missing-David-Sawyer-McFarland/dp/1491918055, which is the location of the resource on the host. This could, for example, be stored in /var/www/amazon/CSS-Missing-David-Sawyer-McFarland/dp/1491918055 if you were running this website on your server.

HTTP Request

An HTTP request is what the web browser uses to communicate with a web server. A request consists of:

HTTP

- Method: such as GET, POST, PUT, DELETE
- URL: path to locate the resource
- Version: usually 1.1, the current version of HTTP
- Headers: a set of headers, each taking the form of a header name and value
- Entity Body: an optional set of data to send to the server

The method controls what type of request it is. GET is used to fetch a resource. POST is used to create a resource. PUT is used to update a resource. DELETE is used to delete a resource.

HTTP GET Request

An example GET request could be:

```
GET /CSS-Missing-David-Sawyer-McFarland/dp/1491918055 HTTP/1.1
Host: www.amazon.com
```

This would fetch the resource /CSS-Missing-David-Sawyer-McFarland/dp/1491918055 from the web server. Since a web server can run more than one site, the Host header specifies that this resource is for the site named www.amazon.com.

HTTP Post Request

An example POST request could be:

```
POST /blog/how-to-make-pizza
Host: homecooking.com

Step 1: Make the dough

4 c flour
1 t sugar
2 1/4 t yeast
2 t sugar
1 1/2 c water
2 T olive oil
...
```

This would tell the web server to create a resource at the path /blog/how-to-make-pizza for the homecooking.com website, with the content listed in the entity body.

HTTP Response Codes

When a browser sends a request to a web server, the server will return a response code indicating whether the request was successful. These are some of the most common response codes you will encounter:

- 200 OK: The request was successful
- 304 Not Modified: The browser already has a copy of the resource and it hasn't changed since it last downloaded it.
- 403 Forbidden: The browser tried to fetch a resource that it doesn't have permission to access. This could be because the resource requires the user to login first, or perhaps the web server doesn't have the file permissions set correctly.
- 404 Not Found: The server could not find the resource the browser is requesting.