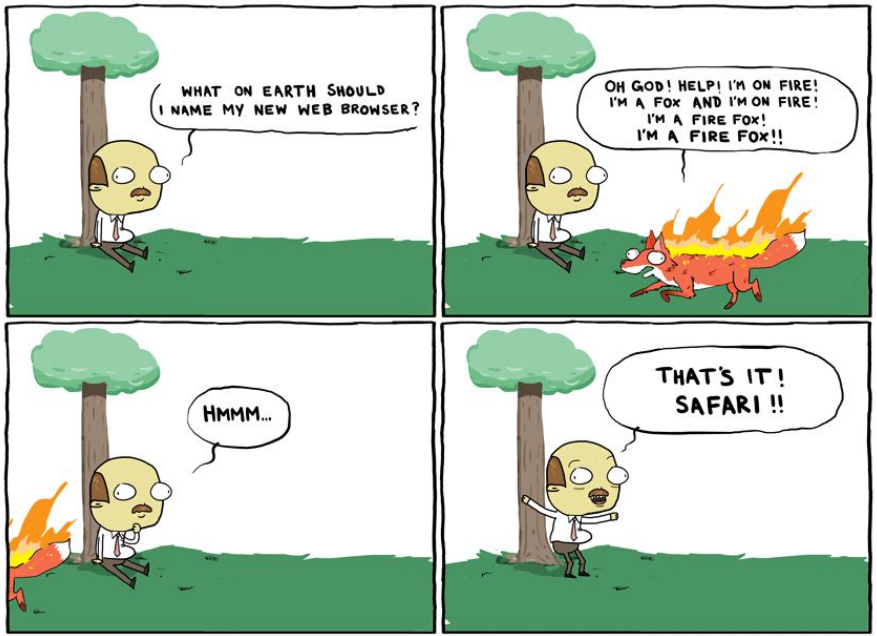


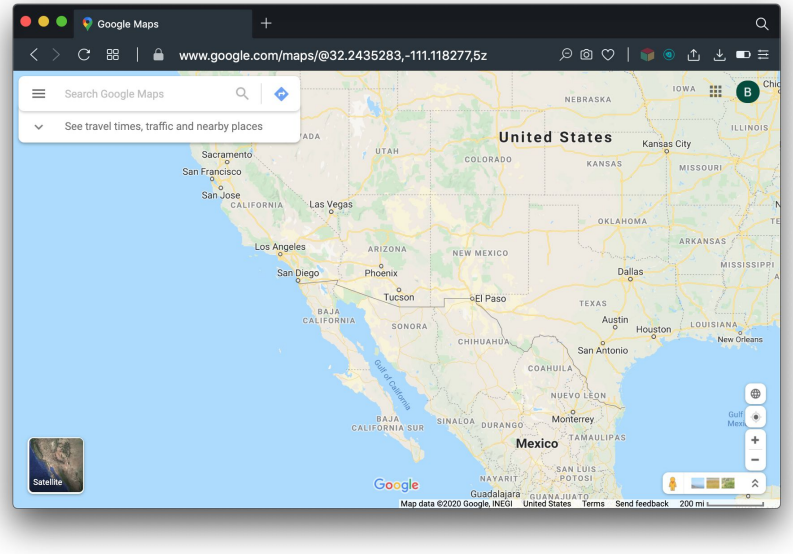
CSc 337 - Browsers, URLs, and HTTP

Benjamin Dicken

What is a web browser?

A web browser ... is a software application for accessing information on the World Wide Web. When a user requests a web page from a particular website, the web browser retrieves the necessary content from a web server and then displays the page on the screen.

(Wikipedia)

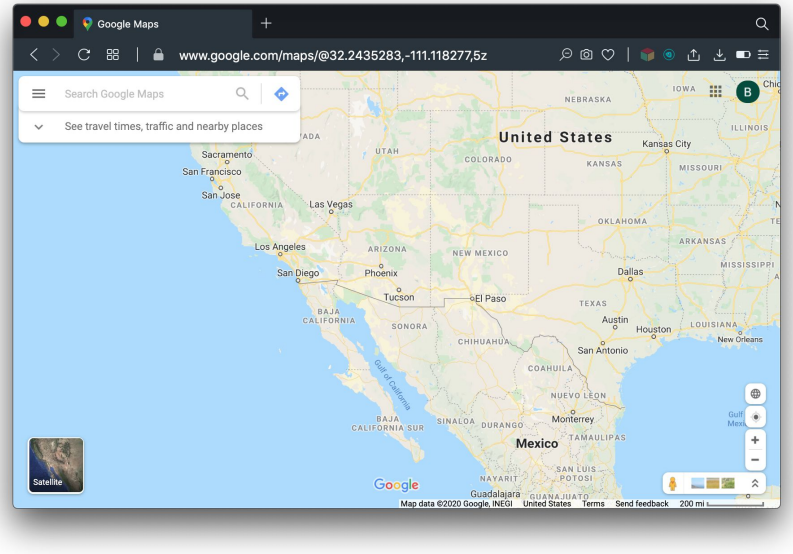


What is a web browser?

Content (pages) are accessed via the Uniform Resource Locator, URL

Can manually request pages via URL

Don’t always have to memorize URLs (search engines, bookmarks, etc)



What do you think are the top three Desktop browsers?

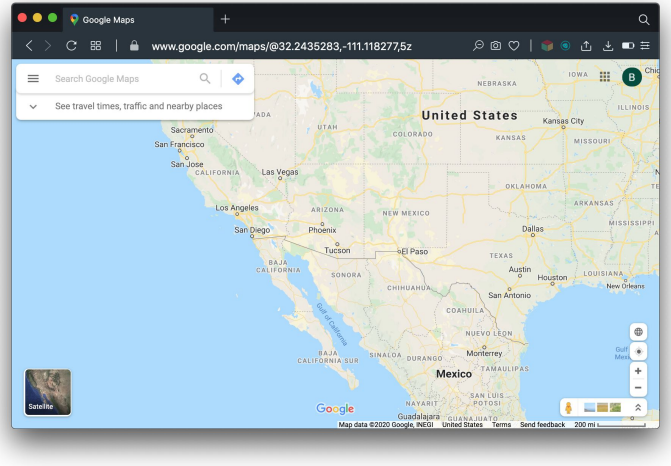
What about for mobile?

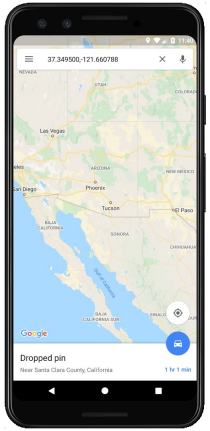
What do you think are the top three Desktop browsers?

https://gs.statcounter.com/browser-market-share

What about for mobile?

https://gs.statcounter.com/browser-market-share/mobile/worldwide





- Access pages/apps via URL
- Page/App code runs within the browser

- Discover apps via the **store**
- Access by touching icon
- Code runs on the operating system

Uniform Resource Locator

https://www.reddit.com/r/UofArizona/

Uniform Resource Locator

The PATH to the specific resource or file within the specified domain.

The DOMAIN - Used to identify the location (computer, server, or group of servers) to get the resource from

The SCHEME (or PROTOCOL). Used to communicate between the browser, and the source of the page (server)

Starting domains with www isn't required, just convention

Domains can end with things other than .com

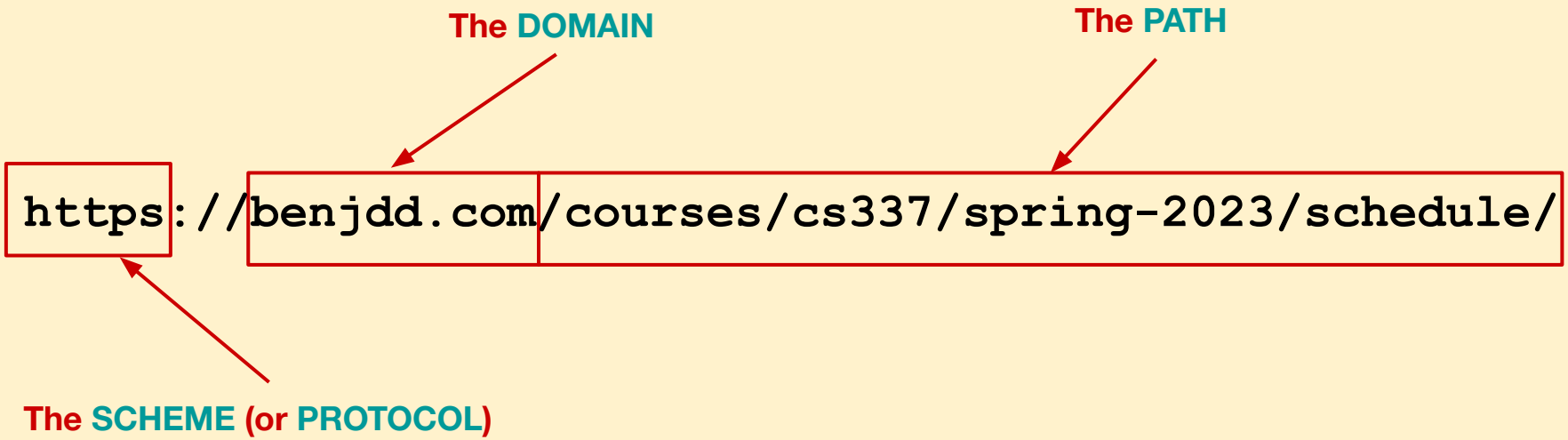
Other stuff can go here, more on that in the future

ICA

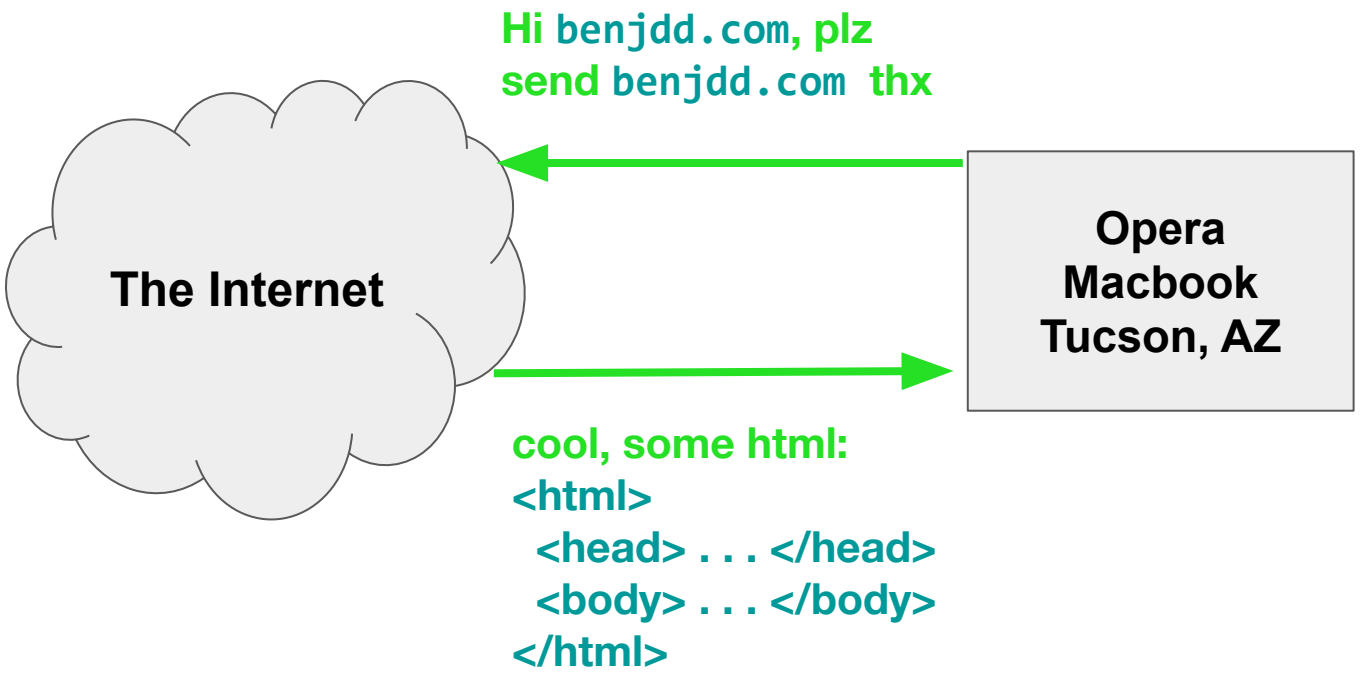
What is the PATH, DOMAIN, and SCHEME for this URL?

`https://benjdd.com/courses/cs337/spring-2023/schedule/`

What is the PATH, DOMATIN, and SCHEME for this URL?



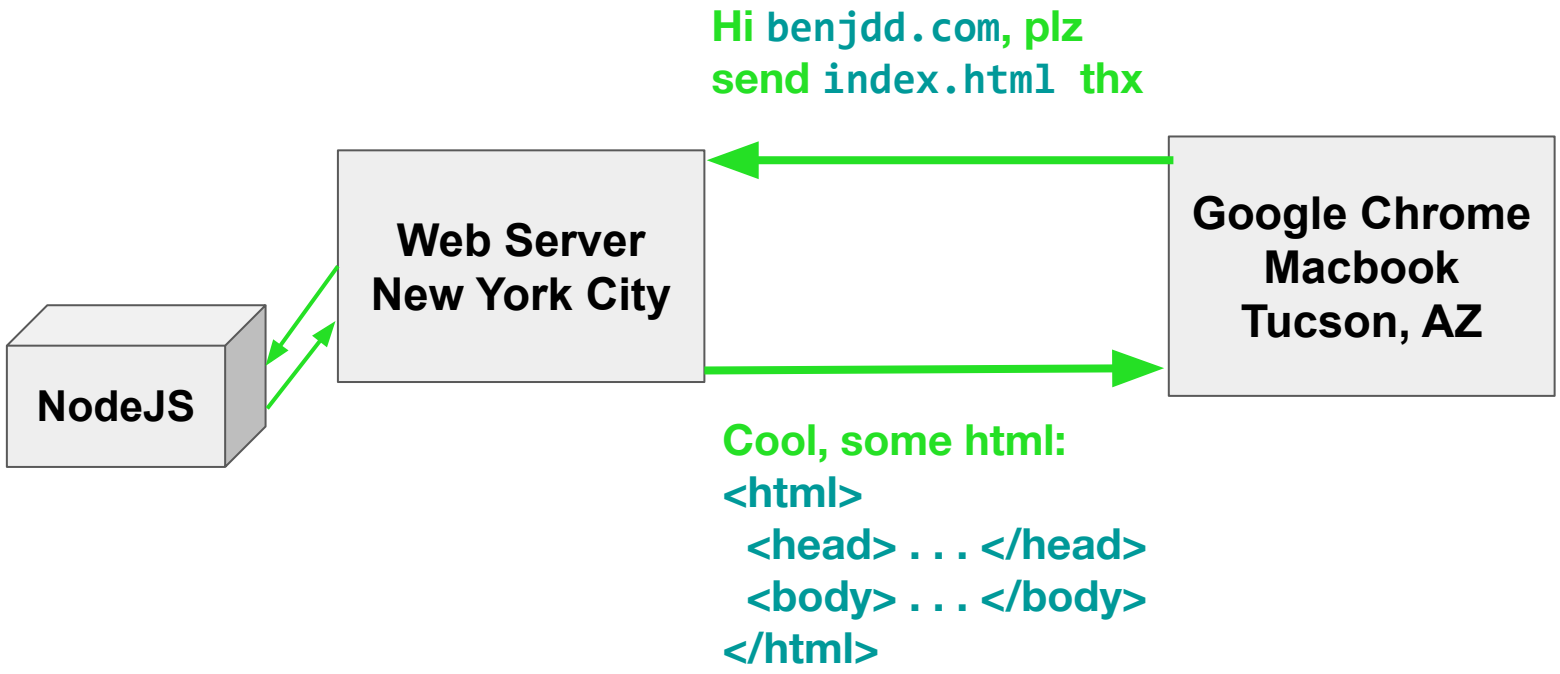
HTTP


Hi benjdd.com, plz
send benjdd.com thx
The Internet
Opera
Macbook
Tucson, AZ
cool, some html:
<html>
<head> . . . </head>
<body> . . . </body>
</html>

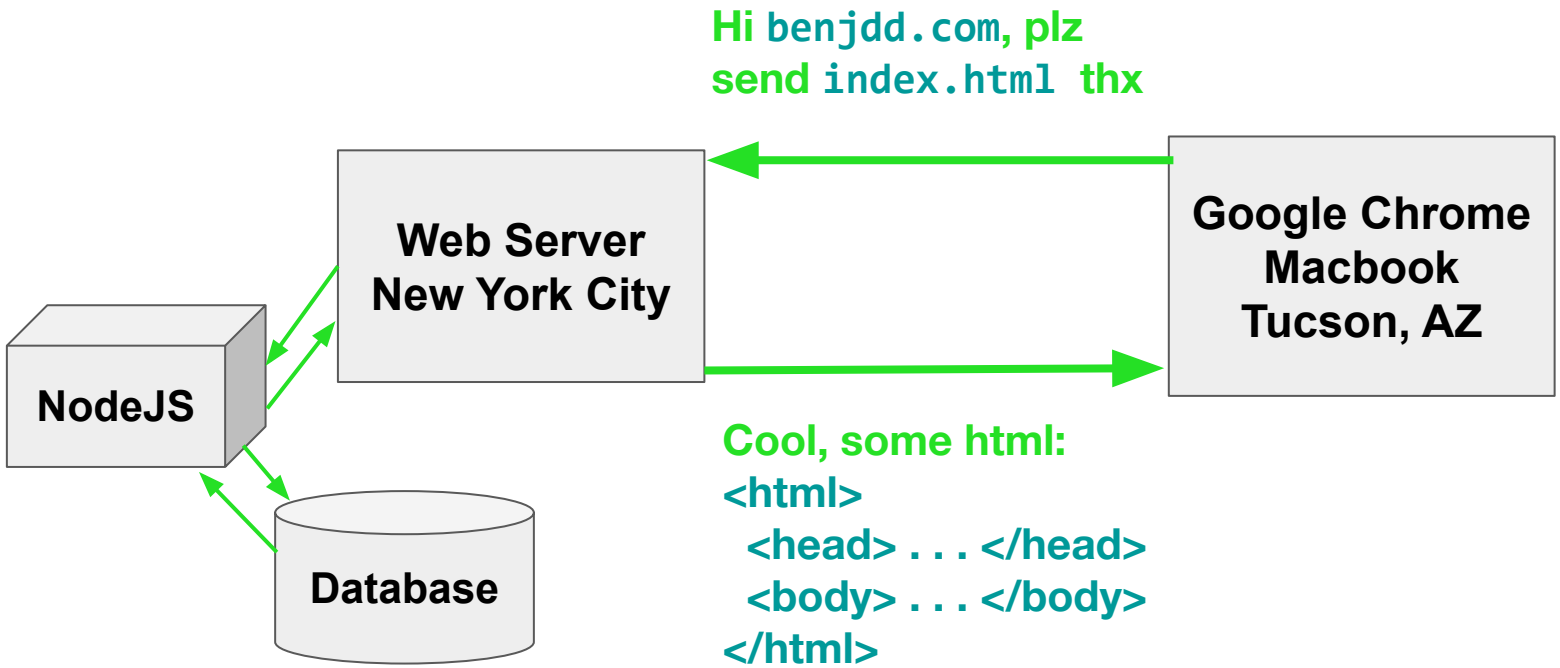
HTTP

- HyperText Transfer Protocol is the standard of communication for web applications
- Used to send data between the client and server
  - Client = Web browser
  - Server = The computer hosting content
- Request-response system

HTTP


Hi benjdd.com, plz
send index.html thx
Web Server
New York City
Google Chrome
Macbook
Tucson, AZ
NodeJS
Cool, some html:
<html>
<head> . . . </head>
<body> . . . </body>
</html>

HTTP

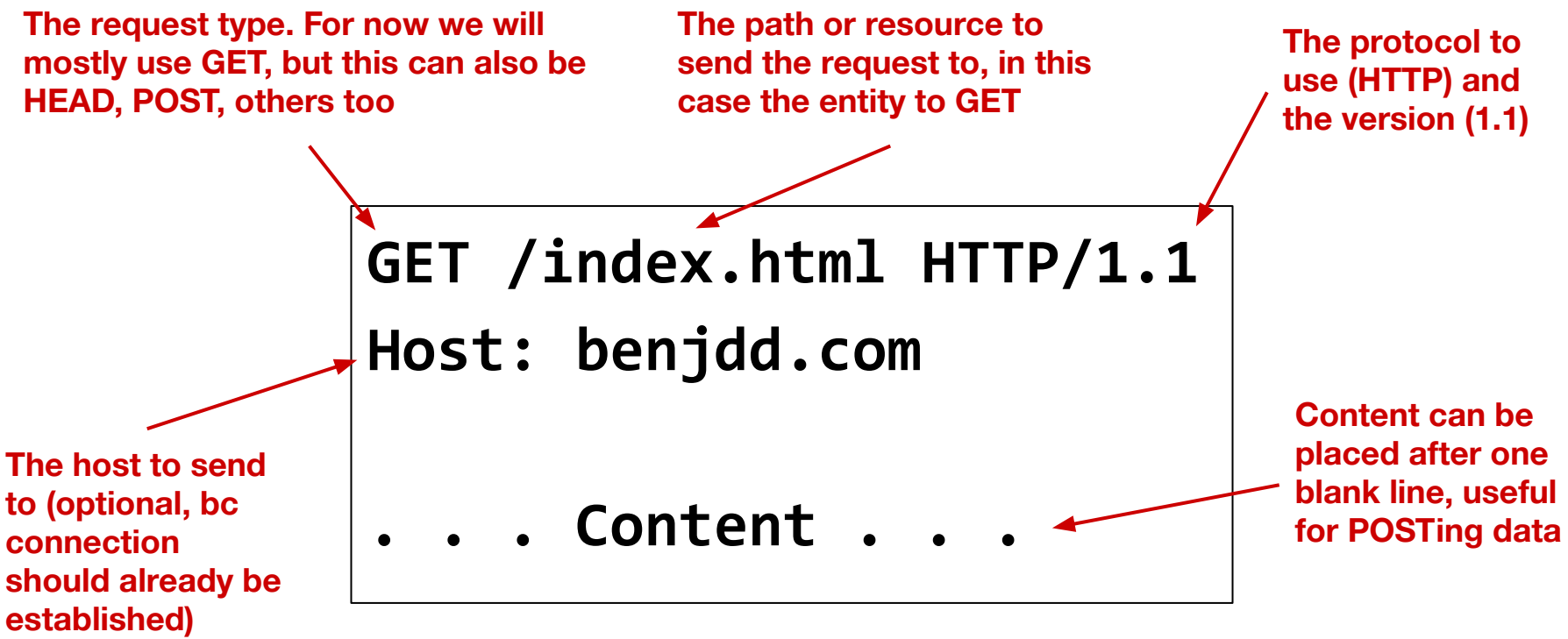

Hi benjdd.com, plz
send index.html thx
Web Server
New York City
Google Chrome
Macbook
Tucson, AZ
NodeJS
Database
Cool, some html:
<html>
<head> . . . </head>
<body> . . . </body>
</html>

HTTP Types of requests

- **GET**: A fetch-only request. Implicitly indicates that there is nothing within the request that the server is expected to remember or store
- **HEAD**: Same as GET, except does not include the actual content, only headers (metadata)
- **POST**: Indicates that the request contents contain something expected to be stored by the server, such as a forum message, youtube comment, etc.

Initially focus on **GET** for this course. **POST** later on.

HTTP Request Format



HTTP Response Format

Response code 200 (means its OK)

A bunch of other headers (metadata)

The contents of the response, perhaps to be rendered and displayed by a browser!

```
HTTP/1.1 200 OK
Accept-Ranges: bytes
Last-Modified: Mon, 18 May 2020 20:52:33 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 62
Date: Mon, 18 May 2020 20:53:25 GMT
Connection: keep-alive

. . . Response Content . . .
```

HTTP

```
GET /hi.html HTTP/1.1
Host: benjdd.com
```


```
HTTP/1.1 200 OK
X-Powered-By: Express
Accept-Ranges: bytes
Cache-Control: public, max-age=0
Last-Modified: Mon, 18 May 2020 20:52:33 GMT
ETag: W/"3e-172298f40a7"
Content-Type: text/html; charset=UTF-8
Content-Length: 62
Date: Mon, 18 May 2020 20:53:25 GMT
Connection: keep-alive

<html>
</head><title>hi</title><head>
<body>hi</body>
</html>
```

HTTP

Say that a website has a feature where a user can post a comment on an image (imagine leaving a comment on an instagram post). On the site, the comment should be displayed along with a timestamp for when it was posted and the users name who posted it.

From the developers perspective, what kind of request should be made to the server to save this message: **GET**, **POST**, or **HEAD** ?

Playing around with HTTP

- Generally your Browser / operating system network code / and the web application handles the details of HTTP
- As a user, just type in the URL you want to go to, hit enter, and the page loads!
- You can use **telnet** to manually submit HTTP requests and **openSSL** for HTTPS

Telnet and OpenSSL

- HTTP request with Telnet:

```
telnet benjdd.com 80
GET /hi.html HTTP/1.1
host: benjdd.com
```

- HTTPS request with OpenSSL:

```
openssl s_client -connect benjdd.com:443
GET /hi.html HTTP/1.1
host: benjdd.com
```

OpenSSL

Windows instructions to install:
https://www.osradar.com/install-openssl-windows/

Mac instructions to install:
https://yasar-yy.medium.com/installing-openssl-library-on-macos-catalina-6777a2e238a6

Status Code

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

https://http.cat

Responding with HTML

- So we talked about requests and responses, but what about the contents?
- For get requests, the contents of the responses are the contents of the site/app
- HTML, CSS, Javascript, etc
- Next up, HTML!

```
HTTP/1.1 200 OK
X-Powered-By: Express
Accept-Ranges: bytes
Cache-Control: public, max-age=0
Last-Modified: Mon, 18 May 2020 20:52:33 GMT
ETag: W/"3e-172298f40a7"
Content-Type: text/html; charset=UTF-8
Content-Length: 62
Date: Mon, 18 May 2020 20:53:25 GMT
Connection: keep-alive

<html>
</head><title>hi</title><head>
<body>hi</body>
</html>
```