Web Request-Response

Web Clients **send** a **web request**

Web Servers **listen** for **web requests**

Web Servers **respond** with **web responses**

Web Clients **receive web responses**

Basic HTTP is 1:1 request:response

- non-HTTP options exist for more
  - websockets, notifications
- Beyond this class

Request/Response Structure

- Request line (request only)
- Status line (response only)
- **Headers**
- Body

We can examine the request/response

- `curl -v URL` will show request/response
- **Browser DevTools** will let us see headers

Request line

The request begins with an **HTTP METHOD**

- Ex: `GET`, `POST`, etc

It then has the **path**

- Ex: `/search?q=smug+cats`, `/students/grades.html`
- Minimum `/`
- Plus any query parameters (part after `?`)
- Not the fragment (after any `#`)

It ends with the **protocol version**

- Ex: `HTTP/1.1`, `HTTP/2`, etc

Request line method

HTTP requests have one of a set list of "**methods**"

- GET
    - No request body and is "idempotent"
    - GET is most common method
- POST
- PUT
- DELETE
- PATCH
- OPTIONS
- TRACE
- HEAD

Request line path

The **path** of the **request** line

- Includes any **query parameters**
  - Ex: `?foo=bar&baz=2`
- Does NOT include any **hash fragment**
  - Ex: `#foo`
  - Fragments are used by the client only

Webserver decides what **response** to give

- Based on **method** + **path**
- **Parameters** usually considered later

Request line protocol version

Most requests are `HTTP/1.1`

- Most of the web only 1 small change from original
- Despite decades of use!

HTTP/2 is out there and growing

- Mostly efficiency improvements

HTTP/3 exists and is being worked on

- More efficiency/performance
- Same underlying concepts

Headers

Headers are text key/value pairs, one per line

Format is:

```
some-header-name: some-header-value
```

Headers are info ABOUT the request

- Date and time
- Size
- Any special authorization information
- Browser information
- Encryption info

Can be seen in your browser DevTools

Body

The contents of the body can be....anything

Decided by any headers that define what to expect

Common options:

- URL-encoded key-value pairs
    - Ex: `foo=bar&baz=my%20cat`
- Structured text data
    - Ex: JSON, XML, etc
- Binary data
    - images, sound, etc

Response status

A web response starts with a line of 3 parts:

- Protocol version (just like end of request line)
- Numeric status code
    - **https://developer.mozilla.org/en-US/docs/Web/HTTP/Status**
    - **https://http.cat/**
- Text message
    - Human readable text for numeric status code

Examine your request/response in the browser

Have a problem in code between client and server?

Always check the request/response

- See which side is sending the wrong thing

Don't waste time solving a non-problem

Summary

- HTTP is a 1:1 series of
  - Client sending request
  - Server getting request
  - Server responding with response
  - Client getting response
- Request:
  - **method**, **path+query**, **headers**, **body** (maybe)
- Response:
  - **status**, **headers**, and optional **body**
- Can see requests/responses in browser DevTools