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(req)/Status line(res)
- Headers
- Body

Request line

The request begins with an **HTTP METHOD**

It then has the **path** (plus any query parameters, but not the fragment)

It ends with the **protocol version**

Request line method

HTTP requests have "methods"

These are one of a set of defined options:

- GET
- POST
- PUT
- DELETE
- PATCH
- OPTION
- TRACE

"GET" has no request body and is "idempotent"

Request line path

The path of the request line

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

Webserver decides how to respond

- Based on method + path

Request line protocol version

Most requests are `HTTP/1.1`

- Despite decades of use, most of the web you know has only involved 1 version change, and that one is small

HTTP/2.0 is out there and growing

- Mostly efficiency improvements

HTTP/3 exists and is being worked on

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**, and optional **body**
- Response: **status**, **headers**, and optional **body**
- Can see requests/responses in browser DevTools