

WEB REQUEST-RESPONSE

web clients issue a **web request**

web servers listen for **web requests**

web servers respond with **web responses**

web clients receive **web responses**

REQUEST/RESPONSE STRUCTURE

- Request line (Request **only**)
- 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"

- GET - used in html forms
- POST - used in html forms
- PUT
- DELETE
- PATCH
- OPTION
- TRACE

"GET" calls have no request body and are idempotent

We'll cover the others later

REQUEST LINE PATH

The path of the request line

- includes any query parameters
 - e.g. `?foo=bar&baz=2`
- does NOT include any fragment
 - e.g. `#foo`
 - fragments are used by the client only

The webserver will decide how to handle the request based on method + path

REQUEST LINE PROTOCOL VERSION

Most requests are `HTTP/1.1`

- Despite decades of use, most of the 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 based key/value pairs, one per line

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

These are ABOUT the request, but not THE request

- Date and time
- size
- any special authorization information
- browser information
- encryption info

This 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 (e.g. `foo=bar&baz=my%20cat`)
- Structured text data (e.g. JSON, XML, etc)
- binary data (images, sound, etc)

RESPONSE STATUS

A web response starts with a line of 3 parts:

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

EXAMINE YOUR REQUEST/RESPONSE IN THE BROWSER

Whenever you have a problem in your code between the client and the server:

First check the request/response to see which side is sending the wrong thing.