Browsers, HTTP

And you!

A brownbag workshop at

by Seth House

@whiteinge
seth@eseth.com

Request / response cycle

A conversation

- Client and server advertise capabilities and supported formats.
- Language, formats, compression, cached resources.

Anatomy of a request

```
GET / HTTP/1.1
Host: localhost:8001

Request body.
```

(Separate with `\r\n`.)

Anatomy of a response

```
HTTP/1.0 200 OK
X-Headers: Here

Response body.
```

Request headers

Request headers

...sent automatically

Request headers

...sent automatically

...not ajax

img, css, script

```
GET /foo.jpg HTTP/1.1
Host: localhost:8001
Referer: http://localhost:8002/
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:72.0) Ge
Accept: image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Pragma: no-cache
Cache-Control: no-cache
Cookie: session_id=8348bd0a15f15e5418e7c0c503e266d9
```

img, css, script

```
GET /foo.jpg HTTP/1.1
Host: localhost:8001
Referer: http://localhost:8002/
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:72.0) Ge
Accept: image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Pragma: no-cache
Cache-Control: no-cache
Cookie: session_id=8348bd0a15f15e5418e7c0c503e266d9
```

(Cookies are sent.)

Non-browser HTTP clients

curl, Ruby, Python, Java, etc send more limited headers:

```
GET / HTTP/1.1
Host: localhost:8001
User-Agent: curl/7.68.0
Accept: */*
```

Non-browser HTTP clients

curl, Ruby, Python, Java, etc send more limited headers:

```
GET / HTTP/1.1
Host: localhost:8001
User-Agent: curl/7.68.0
Accept: */*
```

But they can be specified manually (including cookies):

```
curl localhost:8001 -H 'Accept-Language: en-US'
```

Experiment with an echo server

`echoserver.sh`:

```
#!/usr/bin/env sh
port='8000'
socat -v -T0.05 TCP-L:"$port",reuseaddr,fork \
    system:'cat /path/to/headers/file -'
```

`/path/to/headers/file`:

```
HTTP/1.0 200 OK
Content-Type: text/plain
Connection: close
```

(Must end with a blank line.)

Cookies

Just plain request/response headers.

Cookies

Just plain request/response headers.

First server response:

```
Set-cookie: foo=Foo!
```

Cookies

Just plain request/response headers.

First server response:

```
Set-cookie: foo=Foo!
```

Follow-up requests:

```
Cookie: foo=Foo!
```

Cookies

Just plain request/response headers.

First server response:

```
Set-cookie: foo=Foo!
```

Follow-up requests:

```
Cookie: foo=Foo!
```

The spec defines no specific limit. In practice the widely supported limit is:
50 cookies per domain, 4093 bytes per domain.

Cookies

Expire (near- or far-future):

```
Set-Cookie: foo=Foo!; Expires=Wed, 21 Oct 2021 07:28:00 GMT;
```

Cookies

Expire (near- or far-future):

```
Set-Cookie: foo=Foo!; Expires=Wed, 21 Oct 2021 07:28:00 GMT;
```

Only over HTTPS:

```
Set-Cookie: foo=Foo!; Secure
```

Cookies

Expire (near- or far-future):

```
Set-Cookie: foo=Foo!; Expires=Wed, 21 Oct 2021 07:28:00 GMT;
```

Only over HTTPS:

```
Set-Cookie: foo=Foo!; Secure
```

Restrict JavaScript access:

```
Set-Cookie: foo=Foo!; Secure; HttpOnly
```

Cookies

Expire (near- or far-future):

```
Set-Cookie: foo=Foo!; Expires=Wed, 21 Oct 2021 07:28:00 GMT;
```

Only over HTTPS:

```
Set-Cookie: foo=Foo!; Secure
```

Restrict JavaScript access:

```
Set-Cookie: foo=Foo!; Secure; HttpOnly
```

Include subdomains:

```
Set-Cookie: foo=Foo!; Domain=mx.com
```

Session cookies

Arbitrary key/value pairs defined by the server:

```
Set-Cookie: session_id=abcdef
```

Session cookies

Arbitrary key/value pairs defined by the server:

```
Set-Cookie: session_id=abcdef
```

With some kind of persistent storage (memory, DB, flat files).

Language

```
Accept-Language: en-US,en;q=0.5
```

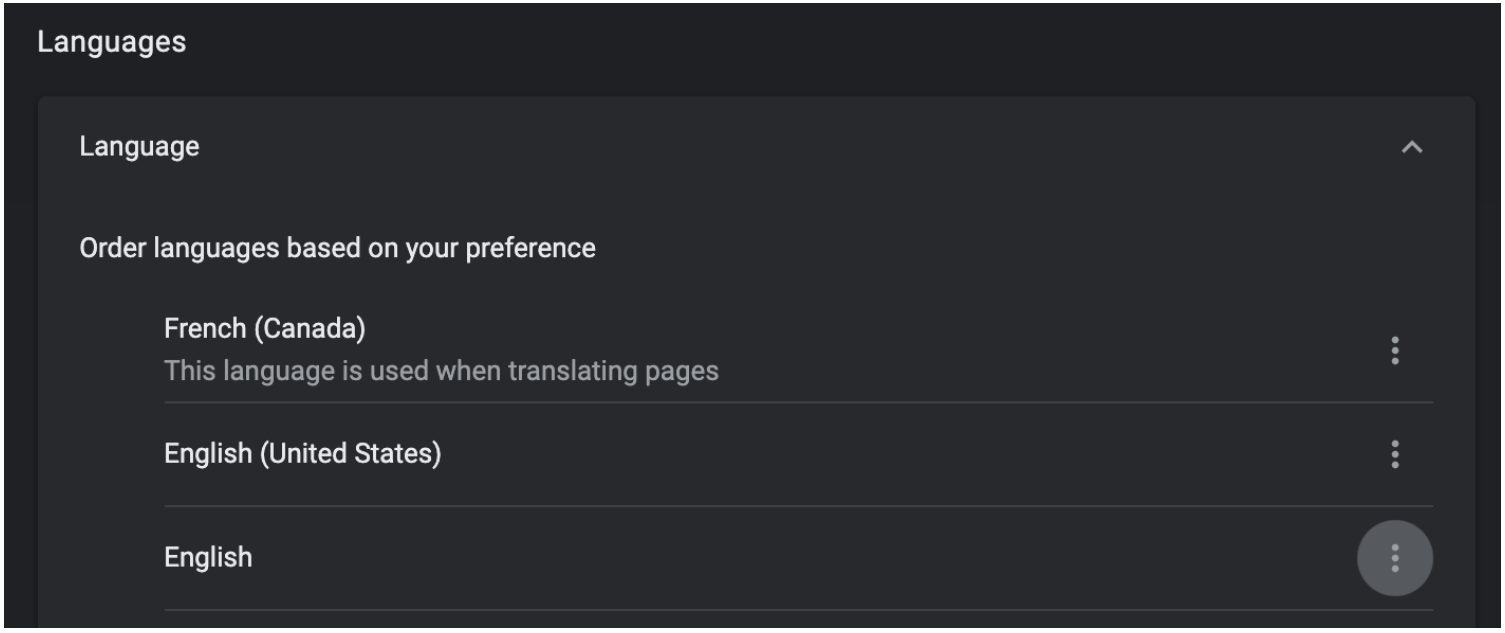
Language

```
Accept-Language: en-US,en;q=0.5
```

Most any browser will send this header:

```
GET / HTTP/1.0
Host: localhost:8001
Accept-Language: en
User-Agent: Lynx/2.8.9rel.1 libwww-FM/2.14 SSL-MM/1.4.1 OpenSSL/1.1.1
```

Language




```
GET / HTTP/1.1
Host: localhost:8001
Accept-Language: fr-CA,fr;q=0.9,en-US;q=0.8,en;q=0.7
```

Content negotiation

```
Accept: */*
```

```
Accept: text/html, application/xhtml+xml, application/xml
```

```
Accept: image/*, image/svg+xml
```

```
Accept: application/json
```

```
Accept: application/x-yaml
```

Mutipart request/response

```
Content-Type: multipart/form-data;boundary="boundaryhere"

--boundaryhere
Content-Disposition: form-data; name="field1"

value1
--boundaryhere
Content-Disposition: form-data; name="field2"; filename="example.txt"

value2
--boundaryhere--
```

HATEOAS

Hypermedia as the Engine of Application State

Compression

```
Accept-Encoding: gzip, deflate
```

Compression

```
Accept-Encoding: gzip, deflate
```

Nginx and Puma (Ruby)

```
|- index.html
|- foo.js
`- foo.js.gz
```

Connection

```
Connection: keep-alive
```

```
Connection: close
```

Host

```
Host: example.com
```

Host

```
Host: example.com
```

Apache vhost config:

```
Listen 80
Listen 192.170.2.1:80

<VirtualHost *:80>
DocumentRoot "/www/example1"
ServerName www.example.com
</VirtualHost>

<VirtualHost *:80>
DocumentRoot "/www/example2"
ServerName www.example.org
</VirtualHost>
```

Specify Host

```
% curl -iSs -H 'Host: microsoft.com' http://example.com
HTTP/1.1 404 Not Found
Content-Type: text/html
Date: Wed, 29 Jan 2020 20:05:02 GMT
Server: ECS (oxr/8313)
Content-Length: 345

<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
         "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
        <head>
                <title>404 - Not Found</title>
        </head>
        <body>
                <h1>404 - Not Found</h1>
        </body>
</html>
```

Specify Host

```
% curl -iSs -H 'Host: microsoft.com' http://example.com
HTTP/1.1 404 Not Found
Content-Type: text/html
Date: Wed, 29 Jan 2020 20:05:02 GMT
Server: ECS (oxr/8313)
Content-Length: 345

<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
         "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
        <head>
                <title>404 - Not Found</title>
        </head>
        <body>
                <h1>404 - Not Found</h1>
        </body>
</html>
```

Connect to specific server behind a load balancer:

```
curl -sS -H 'Host: example.com' http://10.1.1.36:8000
```

User-Agent

```
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:72.0) G
```

User-Agent

```
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:72.0) G
```

- `NCSA_Mosaic/2.0 (Windows 3.1)`

User-Agent

```
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:72.0) G
```

- `NCSA_Mosaic/2.0 (Windows 3.1)`
- "Mosaic Killer"
  `Mozilla/1.0 (Win3.1)`

User-Agent

```
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:72.0) G
```

- `NCSA_Mosaic/2.0 (Windows 3.1)`
- "Mosaic Killer"
  `Mozilla/1.0 (Win3.1)`
- Rebranded as Netscape

User-Agent

```
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:72.0) G
```

- `NCSA_Mosaic/2.0 (Windows 3.1)`
- "Mosaic Killer"
  `Mozilla/1.0 (Win3.1)`
- Rebranded as Netscape
- IE created; impersonated Netscape
  `Mozilla/1.22 (compatible; MSIE 2.0; Windows 95)`

User-Agent

```
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:72.0) G
```

- `NCSA_Mosaic/2.0 (Windows 3.1)`
- "Mosaic Killer"
  `Mozilla/1.0 (Win3.1)`
- Rebranded as Netscape
- IE created; impersonated Netscape
  `Mozilla/1.22 (compatible; MSIE 2.0; Windows 95)`
- Features arms race.

User-Agent

```
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:72.0) G
```

- `NCSA_Mosaic/2.0 (Windows 3.1)`
- "Mosaic Killer"
  `Mozilla/1.0 (Win3.1)`
- Rebranded as Netscape
- IE created; impersonated Netscape
  `Mozilla/1.22 (compatible; MSIE 2.0; Windows 95)`
- Features arms race.
- "And thus Chrome used WebKit, and pretended to be Safari, and WebKit pretended to be KHTML, and KHTML pretended to be Gecko, and all browsers pretended to be Mozilla"

Caching

Response to GET requests: `200`, `301`(permanent), `404`, `206`.

Caching

Response to GET requests: `200`, `301`(permanent), `404`, `206`.

Don't cache:

```
Cache-Control: no-store
```

Caching

Response to GET requests: `200`, `301`(permanent), `404`, `206`.

Don't cache:

```
Cache-Control: no-store
```

Validate before releasing

```
Cache-Control: no-cache
```

Caching

Response to GET requests: `200`, `301`(permanent), `404`, `206`.

Don't cache:

```
Cache-Control: no-store
```

Validate before releasing

```
Cache-Control: no-cache
```

Store in public caching servers, or just browser caches

```
Cache-Control: private
Cache-Control: public
```

Caching

Response to GET requests: `200`, `301`(permanent), `404`, `206`.

Don't cache:

```
Cache-Control: no-store
```

Validate before releasing

```
Cache-Control: no-cache
```

Store in public caching servers, or just browser caches

```
Cache-Control: private
Cache-Control: public
```

Expiration

```
Cache-Control: max-age=31536000
```

Conditional-GET

Request

```
If-Modified-Since: Wed, 21 Oct 2019 07:28:00 GMT
```

Response

```
HTTP/1.0 304 Not Modified
```

Conditional-GET

Request

```
If-Modified-Since: Wed, 21 Oct 2019 07:28:00 GMT
```

Response

```
HTTP/1.0 304 Not Modified
```

Request

```
If-None-Match: "bfc13a64"
```

Response

```
HTTP/1.0 304 Not Modified
```

Conditional-GET

Request

```
If-Modified-Since: Wed, 21 Oct 2019 07:28:00 GMT
```

Response

```
HTTP/1.0 304 Not Modified
```

Request

```
If-None-Match: "bfc13a64"
```

Response

```
HTTP/1.0 304 Not Modified
```

Browsers will sometimes do this automatically.
You can also do this manually via ajax!

Other conditional requests

A conditional POST / PUT

```
If-Unmodified-Since: Wed, 21 Oct 2019 07:28:00 GMT
```

Response

```
HTTP/1.0 412 Precondition Failed
```

Other conditional requests

A conditional POST / PUT

```
If-Unmodified-Since: Wed, 21 Oct 2019 07:28:00 GMT
```

Response

```
HTTP/1.0 412 Precondition Failed
```

`If-Range` to resume downloads or continue following pagination.

Ajax

A tangent...

A tangent...

```
var oReq = new XMLHttpRequest();
oReq.addEventListener('load', function() {
    console.log('XXX', JSON.parse(this.responseText));
});
oReq.oAjaxReq.setRequestHeader('X-MyHeader', 'Foo!')
oReq.open('GET', 'https://api.github.com/users');
oReq.send();
```

A tangent...

```
var oReq = new XMLHttpRequest();
oReq.addEventListener('load', function() {
    console.log('XXX', JSON.parse(this.responseText));
});
oReq.oAjaxReq.setRequestHeader('X-MyHeader', 'Foo!')
oReq.open('GET', 'https://api.github.com/users');
oReq.send();
```

```
fetch('https://api.github.com/users', {
    method: 'GET',
    headers: {'X-MyHeader': 'Foo!'}
})
    .then(resp => resp.json())
    .then(x => console.log('XXX', x));
```

X-Requested-With

```
X-Requested-With: XMLHttpRequest
```

withCredentials

Send cookies!

withCredentials

Send cookies!

```
var oReq = new XMLHttpRequest();
oReq.withCredentials = true;
```

```
fetch('http://example.com', {
    credentials: 'include', // always
    // credentials: 'same-origin',
    // credentials: 'omit', // never
})
```

Forbidden headers

(To JavaScript)

- `Accept-Encoding`
- `Connection`
- `Content-Length`
- `Cookie2`
- `Cookie`
- `Date`
- `Expect`
- `Host`
- `Keep-Alive`
- `Origin`
- `Referer`

Ajax summary

- Just another HTTP client
- Authentication not sent automatically

Response headers

Client requests, server responds accordingly

- `Accept` -> `Content-Type`
- `Accept-Encoding` -> `Content-Encoding`
- `Accept-Language` -> (response body)
- `Cookie` -> `Set-Cookie`
- `If-None-Match` -> `Etag`
- `If-Modified-Since` -> `Last-Modified` / `Date`

Server also declares capabilities and resource info

Resume transfers

- `Accept-Ranges`

Server also declares capabilities and resource info

Resume transfers

- `Accept-Ranges`

Caching

- `X-Cache`
- `Vary`
- `Pragma`

Server also declares capabilities and resource info

Resume transfers

- `Accept-Ranges`

Caching

- `X-Cache`
- `Vary`
- `Pragma`

Proxies & load balancers

- `Forwarded`
- `X-Forwarded-For`
- `Via`
- `Age`


```
Content-Type: text/html; charset=UTF-8
Vary: Accept-Encoding
Content-Encoding: gzip
Server: '; DROP TABLE servertypes; —
Content-Length: 18033
Date: Wed, 15 Aug 2012 13:30:32 GMT
Connection: keep-alive
```

Authentication

Basic access authentication

Request

```
GET / HTTP/1.1
```

Basic access authentication

Request

```
GET / HTTP/1.1
```

Response

```
HTTP 401 Unauthorized
WWW-Authenticate: Basic realm="User Visible Realm"
```

Basic access authentication

Request

```
GET / HTTP/1.1
```

Response

```
HTTP 401 Unauthorized
WWW-Authenticate: Basic realm="User Visible Realm"
```

Request

```
Authorization: Basic QWxhZGRpbjpPcGVuU2VzYW1l
```

Basic access authentication

Request

```
GET / HTTP/1.1
```

Response

```
HTTP 401 Unauthorized
WWW-Authenticate: Basic realm="User Visible Realm"
```

Request

```
Authorization: Basic QWxhZGRpbjpPcGVuU2VzYW1l
```

Built in to most every HTTP server and client.

Digest access authentication

...this space intentionally left blank...

Sessions

Roll-your-own using session cookies.

JSON Web Token (JWT)

...Worth it's own presentation.

OAuth

...Worth it's own presentation.

CORS

What problem are we solving?

Same-origin policy.

Simple CORS

Simple CORS

- Methods:
  - `HEAD`
  - `GET`
  - `POST`

Simple CORS

- Methods:
  - `HEAD`
  - `GET`
  - `POST`
- Headers:
  - `Accept`
  - `Accept-Language`
  - `Content-Language`

Simple CORS

- Methods:
  - `HEAD`
  - `GET`
  - `POST`
- Headers:
  - `Accept`
  - `Accept-Language`
  - `Content-Language`
- `Content-Type`
  - `application/x-www-form-urlencoded`
  - `multipart/form-data`
  - `text/plain`

Enable simple CORS

```
Access-Control-Allow-Origin: http://api.example.com
```

Enable simple CORS

```
Access-Control-Allow-Origin: http://api.example.com
```

```
Access-Control-Allow-Origin: *
```

Enable simple CORS

```
Access-Control-Allow-Origin: http://api.example.com
```

```
Access-Control-Allow-Origin: *
```

```
Access-Control-Allow-Credentials: true
Access-Control-Expose-Headers: X-MyCustom-Header
```

The client only has access to: `Cache-Control`, `Content-Language`, `Content-Type`, `Expires`, `Last-Modified`, `Pragma`

Non-simple CORS

Non-simple CORS

"Preflight request"

```
OPTIONS / HTTP/1.1
Origin: http://api.example.com
Access-Control-Request-Method: PUT
Access-Control-Request-Headers: X-MyCustom-Header
```

Ask the server for permission before making the actual request.

Non-simple CORS

"Preflight request"

```
OPTIONS / HTTP/1.1
Origin: http://api.example.com
Access-Control-Request-Method: PUT
Access-Control-Request-Headers: X-MyCustom-Header
```

Ask the server for permission before making the actual request.

Response

```
Access-Control-Allow-Origin: http://api.example.com
Access-Control-Allow-Methods: GET, POST, PUT
Access-Control-Allow-Headers: X-Custom-Header
```

Two requests for every call

Two requests for every call

```
Access-Control-Max-Age: 31536000
```