</talentlabs>

Express Lecture 1

Introduction to Client and Server Communication

</talentlabs>

Agenda

- Client-Server
- Protocols
- Observes HTTP communication with Chrome

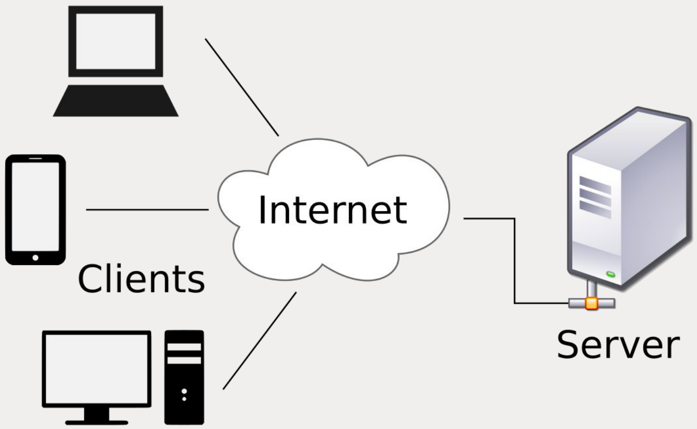
Client-Server

What is a Server

A **server** component **provides a function to one or many clients**. Servers are classified by their functionalities. For example:

- a web server serves web pages and
- a file server serves files.

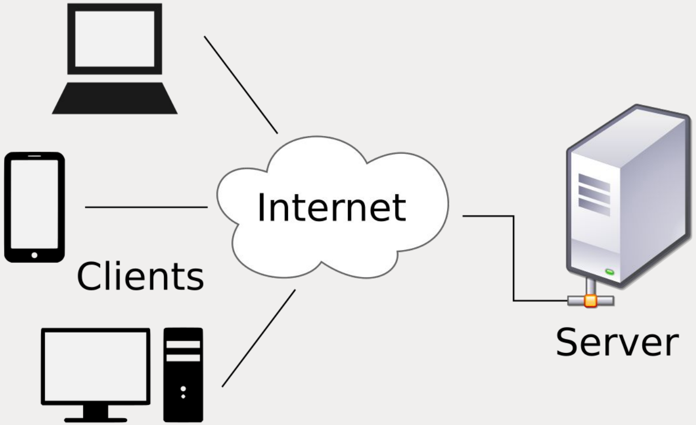
Dropbox UI: Served from a Web server
Dropbox files: Managed by a File server.



What is a Client

Clients are the users of the function provided by a server. They can be

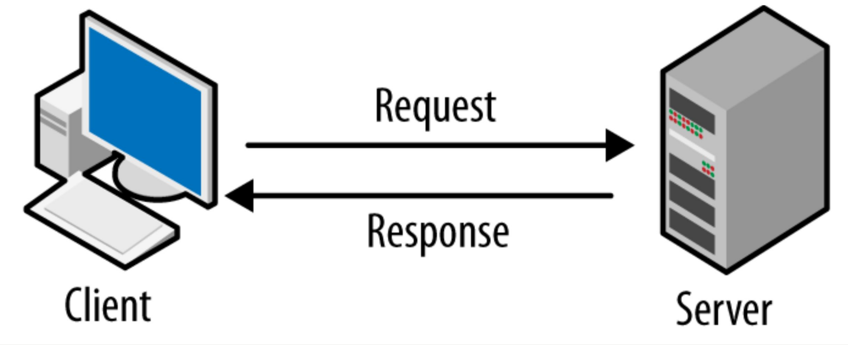
- a computer browser
- a mobile browser
- a mobile application
- an ATM machine
- etc.



Client-Server Communication

Clients and servers **exchange messages in a request–response messaging pattern**. The client sends a request, and the server returns a response.

For example, when a bank customer uses a online banking service with a web browser (the client), the banking web server returns the result to the client web browser for display.



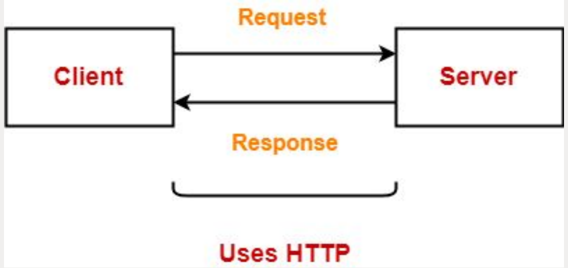
Protocols

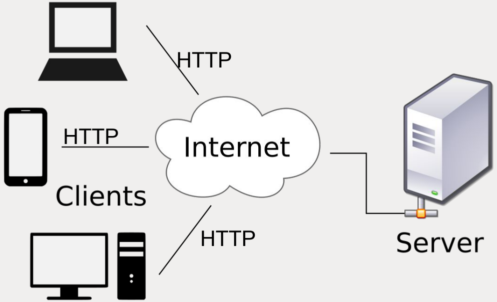
Web Protocol (Common Format)

To communicate, the computers must have a **common format**, and they must follow rules so that both the client and the server know what to expect.

The format and rules of communication are defined in a communications protocol.

A common protocol in modern web technology is the HTTP protocol. That's why people usually talk about HTTP requests and HTTP responses.

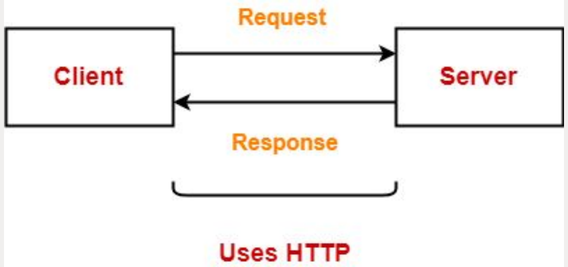


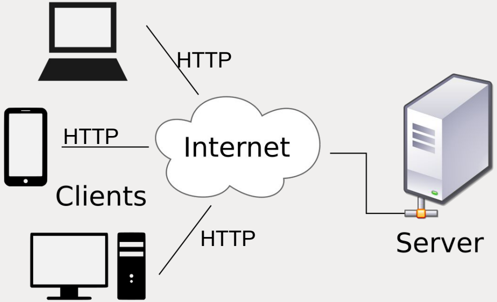


</talentlabs>

Web Protocol (Common Format)

- **HTTP is a protocol which allows the fetching of resources**, such as HTML documents. It is the foundation of any data exchange on the Web and it is a client-server protocol
- Protocol is essential for supporting different client devices (Mobile, Web, IoT etc.)
- As long as they all speak in HTTP, communication is fine.

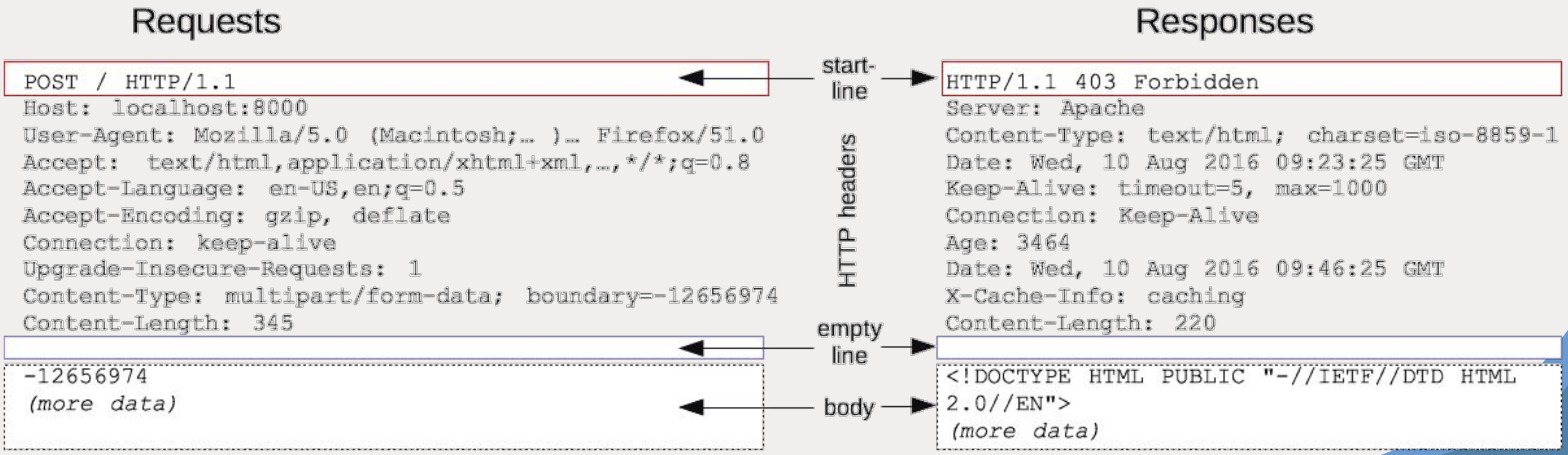




HTTP Message

HTTP 1.1 Message Overview

- HTTP 1.1 Message is **text-based**. Different sections are separated by line breaks.
- **HTTP Request Message and HTTP Response Message** have different formats.
- This section is only for you to better understand the theories. **We don’t need to parse a plain HTTP message, Express will handle it for us.**



Requests

```
POST / HTTP/1.1
Host: localhost:8000
User-Agent: Mozilla/5.0 (Macintosh;… )… Firefox/51.0
Accept: text/html,application/xhtml+xml,…,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data; boundary=-12656974
Content-Length: 345

-12656974
(more data)
```

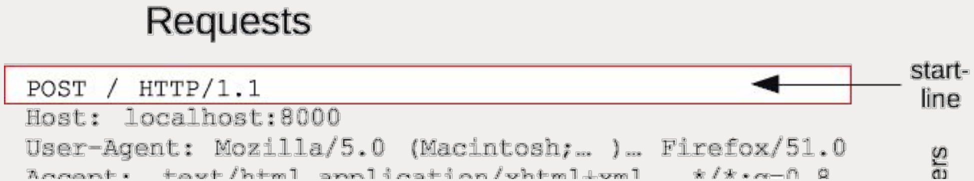
Responses

```
HTTP/1.1 403 Forbidden
Server: Apache
Content-Type: text/html; charset=iso-8859-1
Date: Wed, 10 Aug 2016 09:23:25 GMT
Keep-Alive: timeout=5, max=1000
Connection: Keep-Alive
Age: 3464
Date: Wed, 10 Aug 2016 09:46:25 GMT
X-Cache-Info: caching
Content-Length: 220

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML
2.0//EN">
(more data)
```

</talentlabs>

HTTP 1.1 Request Start line

- HTTP requests are messages sent by the client to initiate an action on the server. Their start-line contain three elements:
  - HTTP method, like **GET**. (Will talk more about this in the the future)
  - The HTTP version, like **HTTP/1.1**
  - The request target, usually the URL/path, like ***/***



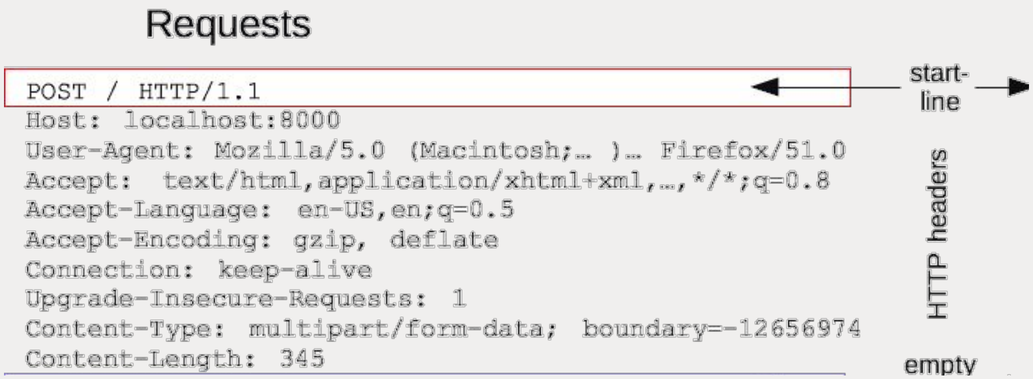
```
POST / HTTP/1.1
Host: localhost:8000
User-Agent: Mozilla/5.0 (Macintosh;… )… Firefox/51.0
Accept: text/html,application/xhtml+xml,…*/*;q=0.8
```

HTTP 1.1 Request Start line

- HTTP requests are messages sent by the client to initiate an action on the server. Their start-line contain three elements:
  - HTTP method, like **GET**. (Will talk more about this in the the future)
  - The HTTP version, like **HTTP/1.1**. (Consider it as a standard version nowadays)
  - The **request target**, usually the URL/path, like ***/***. (Today focus)
    - **Tell the server what do you want to get**, for example
      - /background.png
        - Get an image from the server
      - /anypage.html
        - Get a HTML file from the server

HTTP 1.1 Request Headers

- HTTP Request headers are some key-value pairs to tell more about the HTTP message.
  - Host: It is like **the address to the server**.
  - Content-type: It tells **what kind of data is inside the HTTP body**.
  - Many more others...

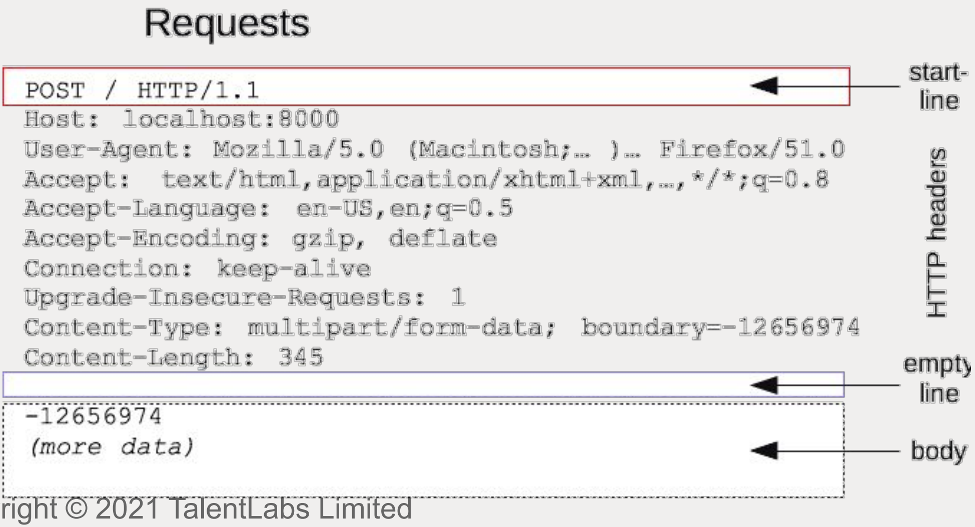


Requests

```
POST / HTTP/1.1
Host: localhost:8000
User-Agent: Mozilla/5.0 (Macintosh;… )… Firefox/51.0
Accept: text/html,application/xhtml+xml,…,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data; boundary=-12656974
Content-Length: 345
```

HTTP 1.1 Request Body

- The message body contains the **actual data the client want to send**.
- You can put different data format in the HTTP request body. Some common examples are:
  - application/json
  - multipart/form-data



```
POST / HTTP/1.1
Host: localhost:8000
User-Agent: Mozilla/5.0 (Macintosh;… )… Firefox/51.0
Accept: text/html,application/xhtml+xml,…,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data; boundary=-12656974
Content-Length: 345

-12656974
(more data)
```

Simple HTTP Request Example

There are many concepts about HTTP, let's take a look at this simple example request:

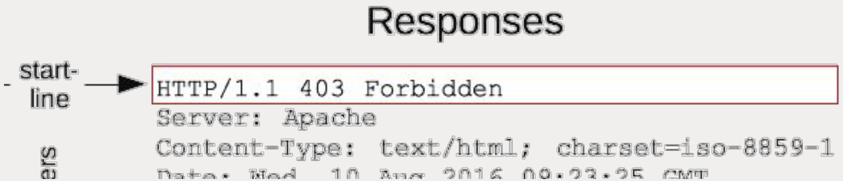
```
GET /about HTTP1.1
Host: www.talentlabs.org
```

The above HTTP Request

1. Asks the backend application located at www.talentlabs.org to ...
2. Gets HTML of the /about page.
3. Contains an empty body, as no additional information is needed here.

HTTP 1.1 Response Status line

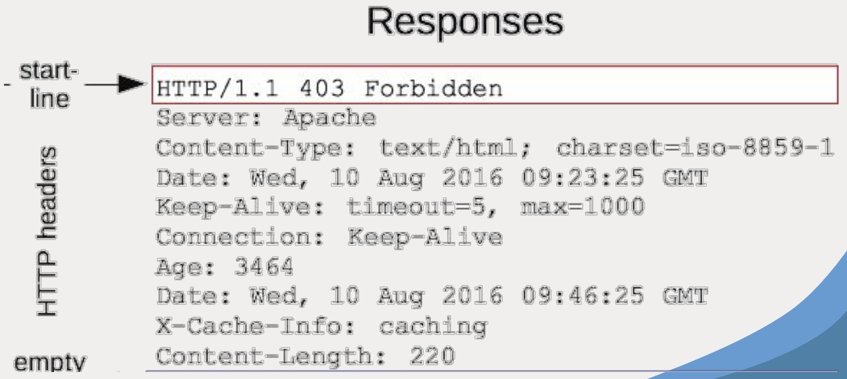
- The start line of an HTTP response, called the status line, contains the following information:
  - The HTTP version, like HTTP/1.1.
  - The status code, like 403. (Will talk more about this in the the future)
  - The text version of the status code, like Forbidden. (Will talk more about this in the the future)



```
HTTP/1.1 403 Forbidden
Server: Apache
Content-Type: text/html; charset=iso-8859-1
Date: Wed, 10 Aug 2016 09:23:25 GMT
```

HTTP 1.1 Response Headers

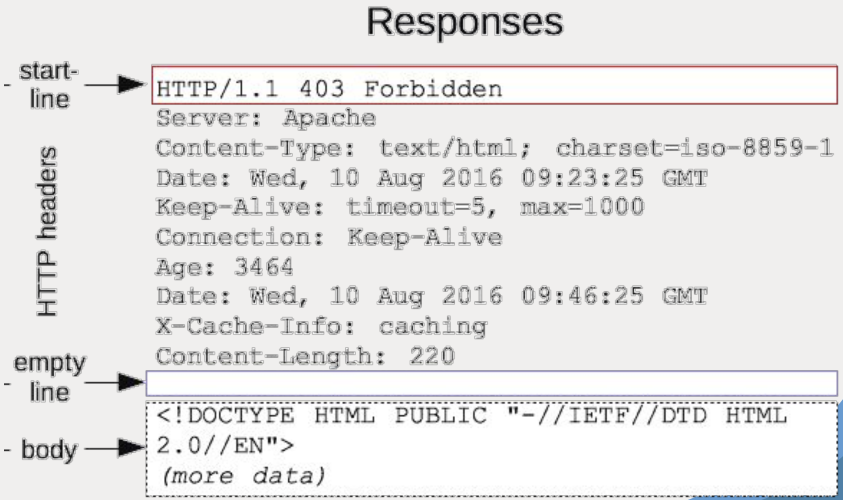
- Similarly, HTTP response headers are some key-value pairs to tell more about the HTTP message.
  - Content-Type: what kind of data is inside the HTTP response body.



```
HTTP/1.1 403 Forbidden
Server: Apache
Content-Type: text/html; charset=iso-8859-1
Date: Wed, 10 Aug 2016 09:23:25 GMT
Keep-Alive: timeout=5, max=1000
Connection: Keep-Alive
Age: 3464
Date: Wed, 10 Aug 2016 09:46:25 GMT
X-Cache-Info: caching
Content-Length: 220
```

HTTP 1.1 Response Body

- The message body contain the **actual data the server want to reply**.
- You can put different data format in the HTTP response body. Some common examples are:
    - application/json
    - text/html



```
HTTP/1.1 403 Forbidden
Server: Apache
Content-Type: text/html; charset=iso-8859-1
Date: Wed, 10 Aug 2016 09:23:25 GMT
Keep-Alive: timeout=5, max=1000
Connection: Keep-Alive
Age: 3464
Date: Wed, 10 Aug 2016 09:46:25 GMT
X-Cache-Info: caching
Content-Length: 220

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML
2.0//EN">
(more data)
```

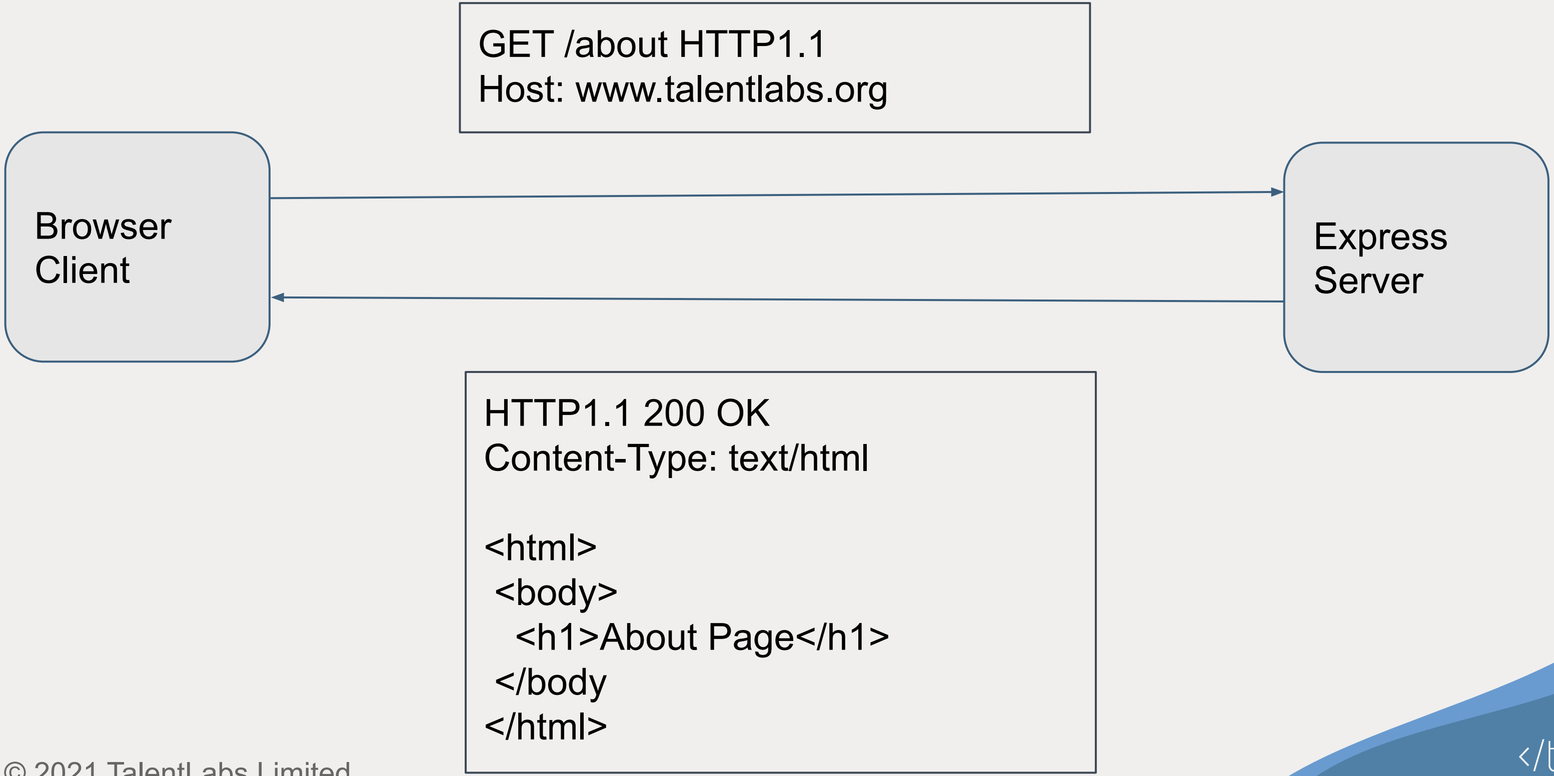
Simple HTTP Response Example

There are many concepts about HTTP, let’s take a look at this simple example request:

```
HTTP1.1 200 OK
Content-Type: text/html

<html>
 <body>
   <h1>About Page</h1>
 </body
</html>
```

The above HTTP response is to reply the client with HTML text the client asked for.

All together



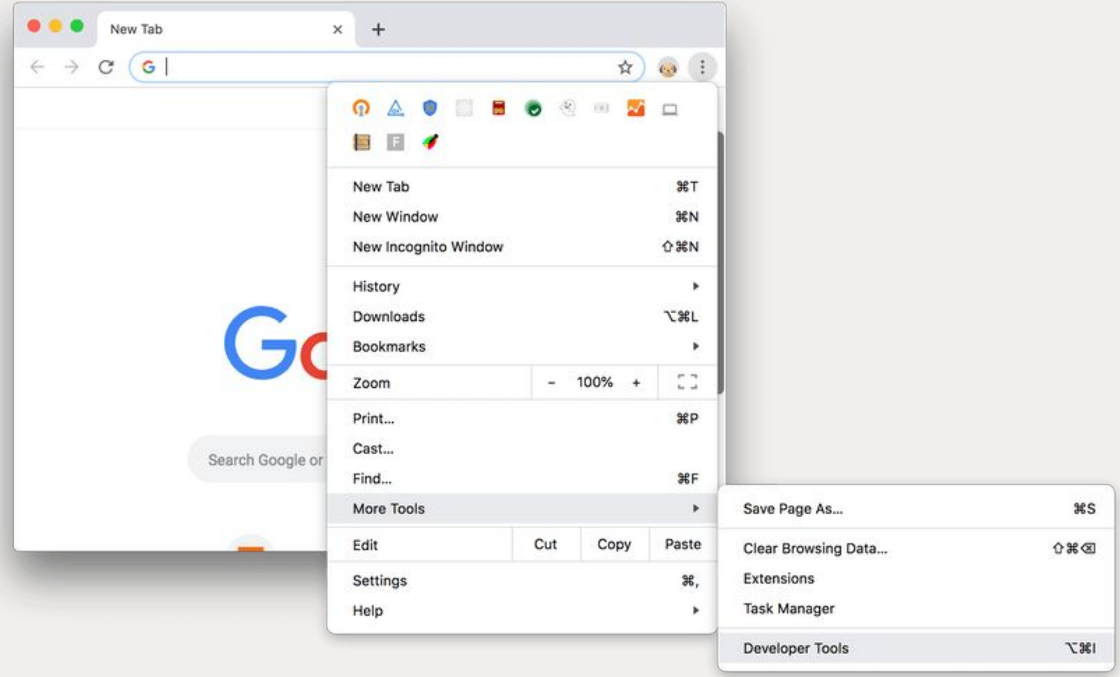
```
GET /about HTTP1.1
Host: www.talentlabs.org
```

```
HTTP1.1 200 OK
Content-Type: text/html

<html>
 <body>
   <h1>About Page</h1>
 </body
</html>
```

Observes HTTP communications with Chrome

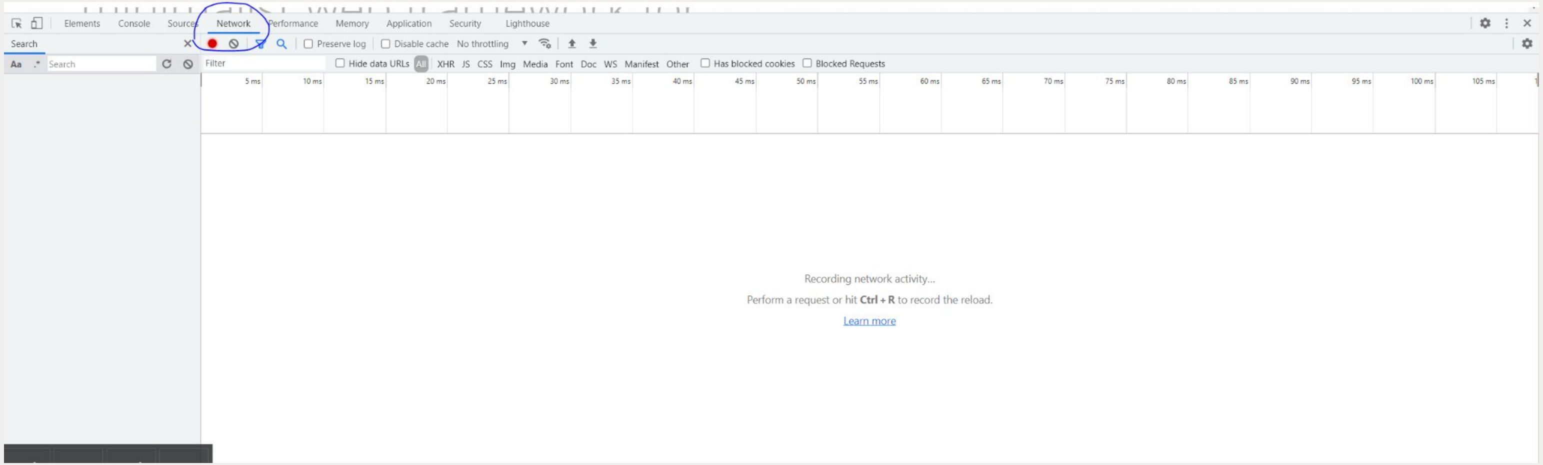
</talentlabs>

Open Chrome Developer Console



</talentlabs>

Visit https://expressjs.com/, Open Chrome Developer Console - Network tab

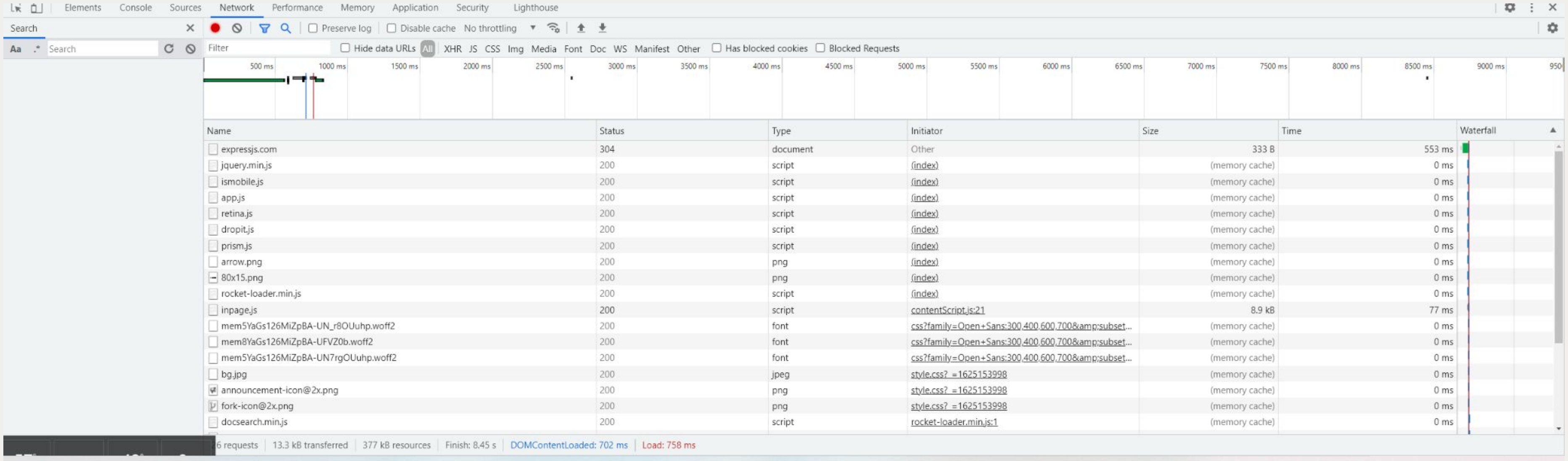


Recording network activity...

Perform a request or hit **Ctrl + R** to record the reload.

Learn more

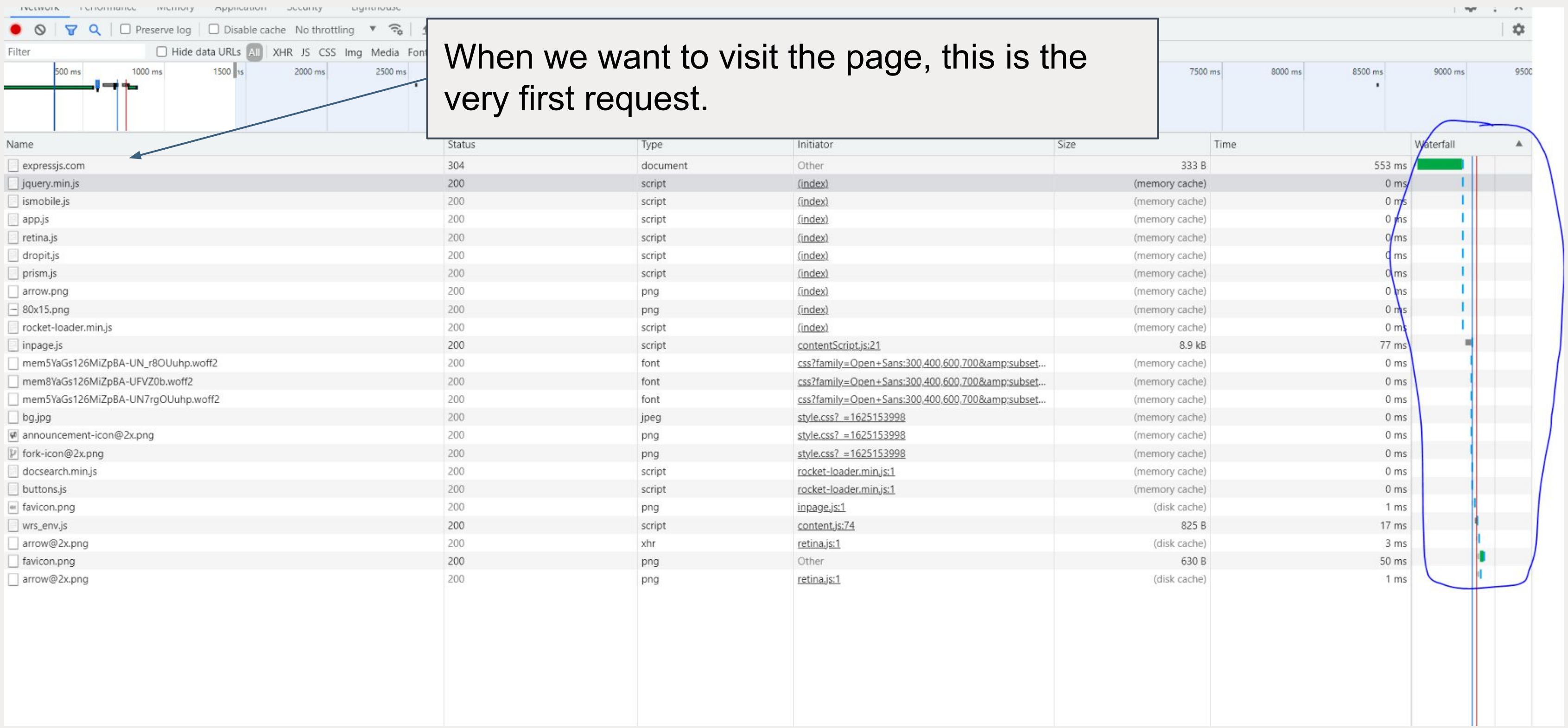
Refresh the page to record the network usage



Name	Status	Type	Initiator	Size	Time	Waterfall
expressjs.com	304	document	Other	333 B	553 ms	
jquery.min.js	200	script	(index)	(memory cache)	0 ms	
ismobile.js	200	script	(index)	(memory cache)	0 ms	
app.js	200	script	(index)	(memory cache)	0 ms	
retina.js	200	script	(index)	(memory cache)	0 ms	
dropit.js	200	script	(index)	(memory cache)	0 ms	
prism.js	200	script	(index)	(memory cache)	0 ms	
arrow.png	200	png	(index)	(memory cache)	0 ms	
80x15.png	200	png	(index)	(memory cache)	0 ms	
rocket-loader.min.js	200	script	(index)	(memory cache)	0 ms	
inpage.js	200	script	contentScript.js:21	8.9 kB	77 ms	
mem5YaGs126MiZpBA-UN_r8OUuhp.woff2	200	font	css?family=Open+Sans:300,400,600,700&amp;subset…	(memory cache)	0 ms	
mem8YaGs126MiZpBA-UFVZ0b.woff2	200	font	css?family=Open+Sans:300,400,600,700&amp;subset…	(memory cache)	0 ms	
mem5YaGs126MiZpBA-UN7rgOUuhp.woff2	200	font	css?family=Open+Sans:300,400,600,700&amp;subset…	(memory cache)	0 ms	
bg.jpg	200	jpeg	style.css?_=1625153998	(memory cache)	0 ms	
announcement-icon@2x.png	200	png	style.css?_=1625153998	(memory cache)	0 ms	
fork-icon@2x.png	200	png	style.css?_=1625153998	(memory cache)	0 ms	
docsearch.min.js	200	script	rocket-loader.min.js:1	(memory cache)	0 ms	

6 requests | 13.3 kB transferred | 377 kB resources | Finish: 8.45 s | DOMContentLoaded: 702 ms | Load: 758 ms

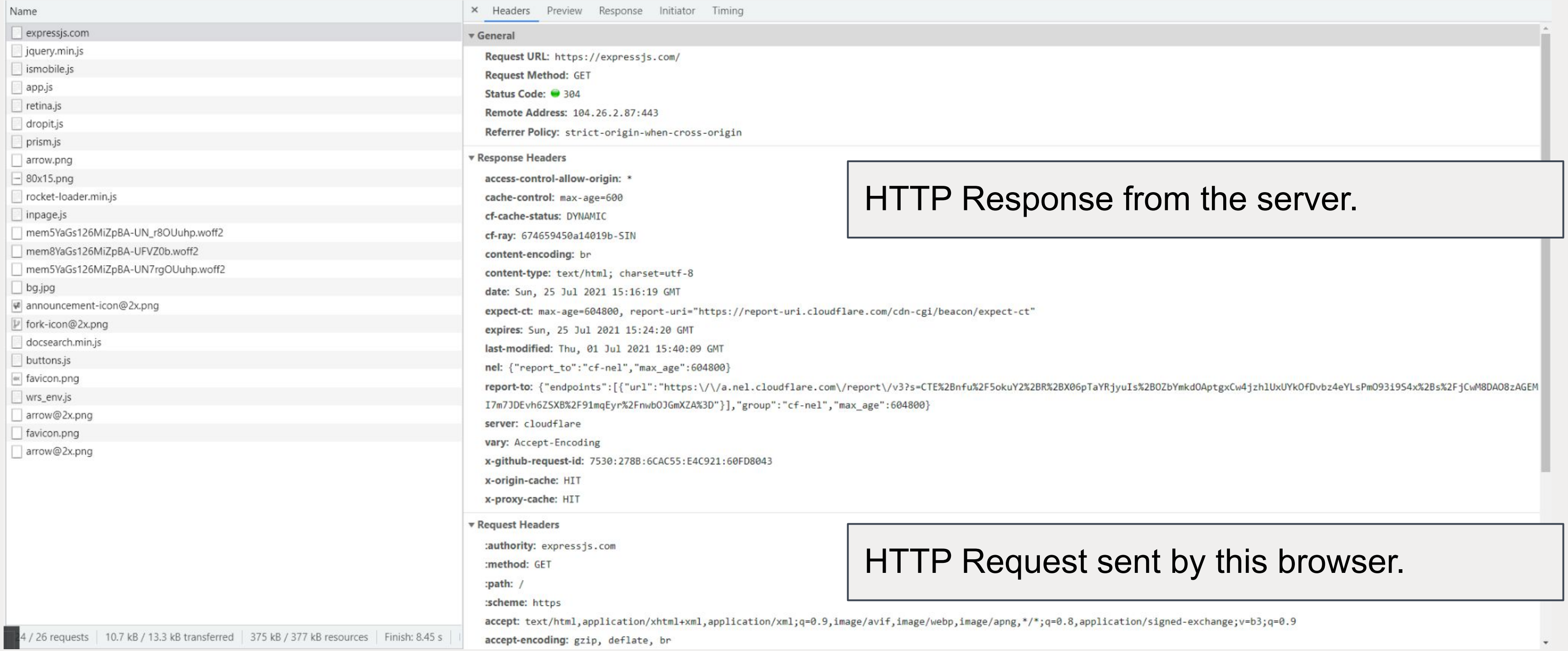
Click the Waterfall column to sort the requests (Sort by request order)



When we want to visit the page, this is the very first request.

Name	Status	Type	Initiator	Size	Time	Waterfall ▲
expressjs.com	304	document	Other	333 B	553 ms	
jquery.min.js	200	script	(index)	(memory cache)	0 ms	
ismobile.js	200	script	(index)	(memory cache)	0 ms	
app.js	200	script	(index)	(memory cache)	0 ms	
retina.js	200	script	(index)	(memory cache)	0 ms	
dropit.js	200	script	(index)	(memory cache)	0 ms	
prism.js	200	script	(index)	(memory cache)	0 ms	
arrow.png	200	png	(index)	(memory cache)	0 ms	
80x15.png	200	png	(index)	(memory cache)	0 ms	
rocket-loader.min.js	200	script	(index)	(memory cache)	0 ms	
inpage.js	200	script	contentScript.js:21	8.9 kB	77 ms	
mem5YaGs126MiZpBA-UN_r8OUuhp.woff2	200	font	css?family=Open+Sans:300,400,600,700&amp;subset...	(memory cache)	0 ms	
mem8YaGs126MiZpBA-UFVZ0b.woff2	200	font	css?family=Open+Sans:300,400,600,700&amp;subset...	(memory cache)	0 ms	
mem5YaGs126MiZpBA-UN7rgOUuhp.woff2	200	font	css?family=Open+Sans:300,400,600,700&amp;subset...	(memory cache)	0 ms	
bg.jpg	200	jpeg	style.css? =1625153998	(memory cache)	0 ms	
announcement-icon@2x.png	200	png	style.css? =1625153998	(memory cache)	0 ms	
fork-icon@2x.png	200	png	style.css? =1625153998	(memory cache)	0 ms	
docsearch.min.js	200	script	rocket-loader.min.js:1	(memory cache)	0 ms	
buttons.js	200	script	rocket-loader.min.js:1	(memory cache)	0 ms	
favicon.png	200	png	inpage.js:1	(disk cache)	1 ms	
wrs_env.js	200	script	content.js:74	825 B	17 ms	
arrow@2x.png	200	xhr	retina.js:1	(disk cache)	3 ms	
favicon.png	200	png	Other	630 B	50 ms	
arrow@2x.png	200	png	retina.js:1	(disk cache)	1 ms	

Click the first request and observe



Name

- expressjs.com
- jquery.min.js
- ismobile.js
- app.js
- retina.js
- dropit.js
- prism.js
- arrow.png
- 80x15.png
- rocket-loader.min.js
- inpage.js
- mem5YaGs126MiZpBA-UN_r8OUuhp.woff2
- mem8YaGs126MiZpBA-UFVZ0b.woff2
- mem5YaGs126MiZpBA-UN7rgOUuhp.woff2
- bg.jpg
- announcement-icon@2x.png
- fork-icon@2x.png
- docsearch.min.js
- buttons.js
- favicon.png
- wrs_env.js
- arrow@2x.png
- favicon.png
- arrow@2x.png

[redacted]4 / 26 requests | 10.7 kB / 13.3 kB transferred | 375 kB / 377 kB resources | Finish: 8.45 s

▾ General

```
Request URL: https://expressjs.com/
Request Method: GET
Status Code: ● 304
Remote Address: 104.26.2.87:443
Referrer Policy: strict-origin-when-cross-origin
```

▾ Response Headers

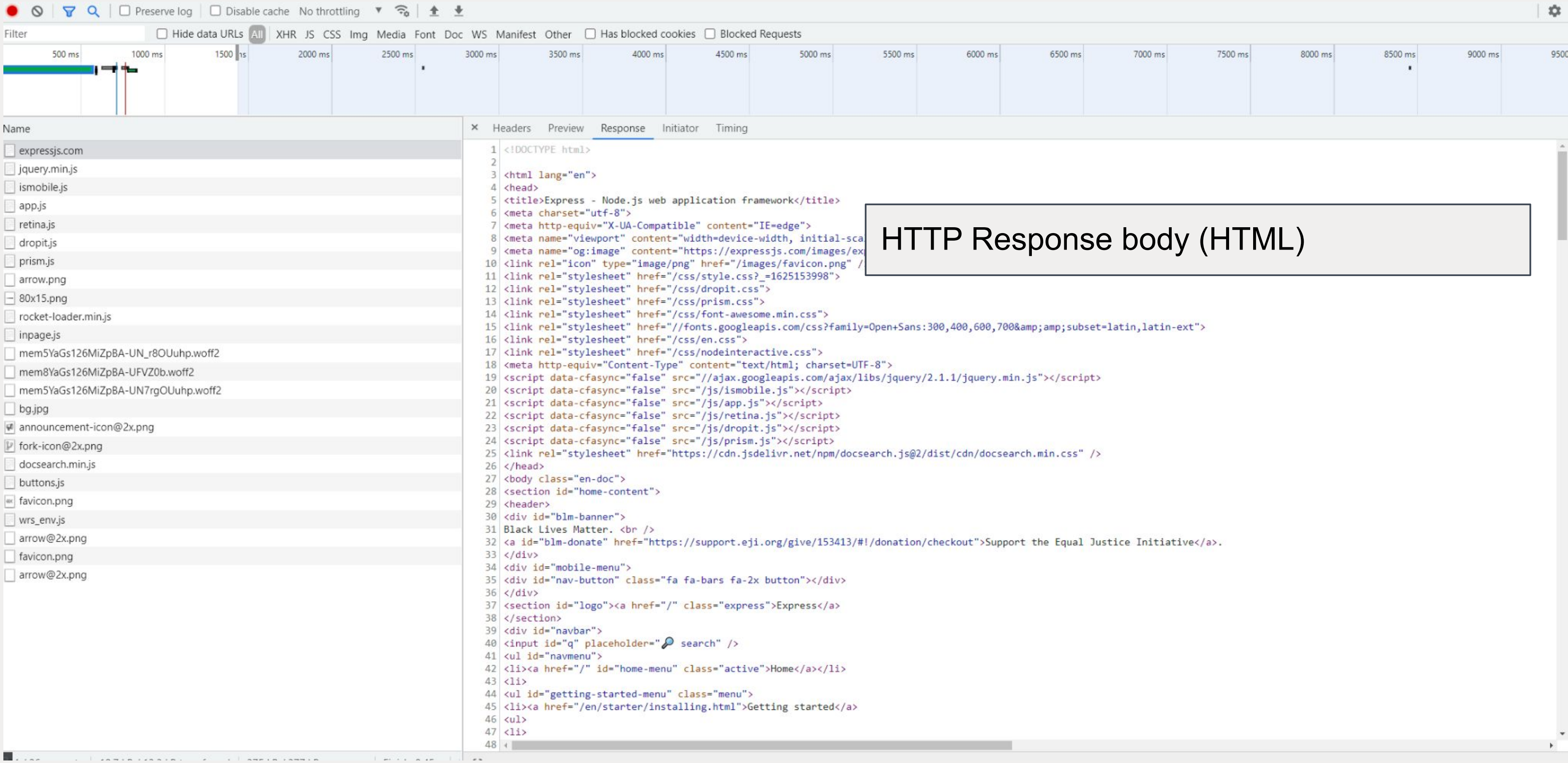
```
access-control-allow-origin: *
cache-control: max-age=600
cf-cache-status: DYNAMIC
cf-ray: 674659450a14019b-SIN
content-encoding: br
content-type: text/html; charset=utf-8
date: Sun, 25 Jul 2021 15:16:19 GMT
expect-ct: max-age=604800, report-uri="https://report-uri.cloudflare.com/cdn-cgi/beacon/expect-ct"
expires: Sun, 25 Jul 2021 15:24:20 GMT
last-modified: Thu, 01 Jul 2021 15:40:09 GMT
nel: {"report_to":"cf-nel","max_age":604800}
report-to: {"endpoints":[{"url":"https:\/\/a.nel.cloudflare.com\/report\/v3?s=CTE%2Bnfu%2F5okuY2%2BR%2BX06pTaYRjyuIs%2BOZbYmkdOAptgxCw4jzhlUxUYkOfDvbz4eYLsPmO93i9S4x%2Bs%2FjCwM8DAO8zAGEM
I7m7JDEvh6ZSXB%2F91mqEyr%2FnwbOJGmXZA%3D"}],"group":"cf-nel","max_age":604800}
server: cloudflare
vary: Accept-Encoding
x-github-request-id: 7530:278B:6CAC55:E4C921:60FD8043
x-origin-cache: HIT
x-proxy-cache: HIT
```

HTTP Response from the server.

▾ Request Headers

```
:authority: expressjs.com
:method: GET
:path: /
:scheme: https
accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
accept-encoding: gzip, deflate, br
```

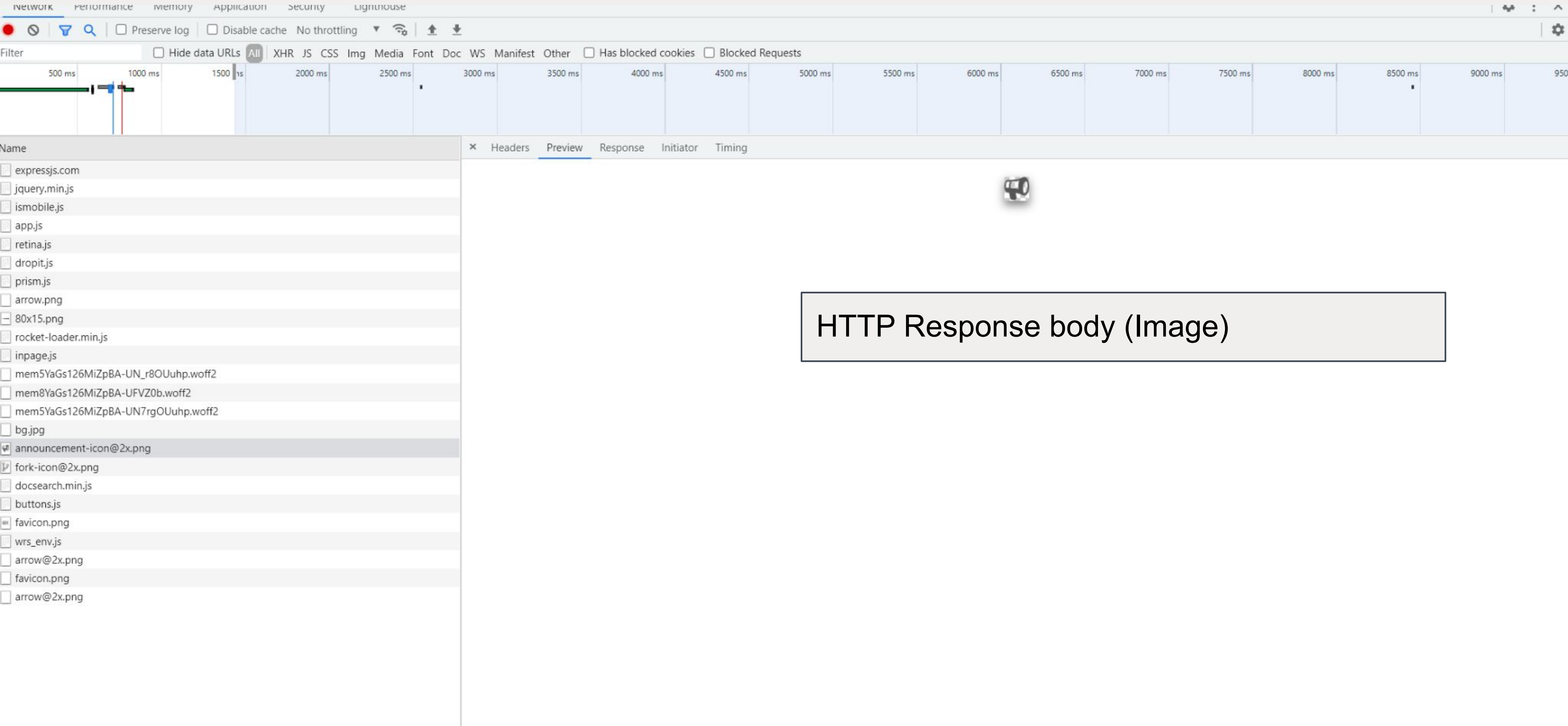
HTTP Request sent by this browser.



```
expressjs.com
jquery.min.js
ismobile.js
app.js
retina.js
dropit.js
prism.js
arrow.png
80x15.png
rocket-loader.min.js
inpage.js
mem5YaGs126MiZpBA-UN_r8OUuhp.woff2
mem8YaGs126MiZpBA-UFVZ0b.woff2
mem5YaGs126MiZpBA-UN7rgOUuhp.woff2
bg.jpg
announcement-icon@2x.png
fork-icon@2x.png
docsearch.min.js
buttons.js
favicon.png
wrs_env.js
arrow@2x.png
favicon.png
arrow@2x.png
```

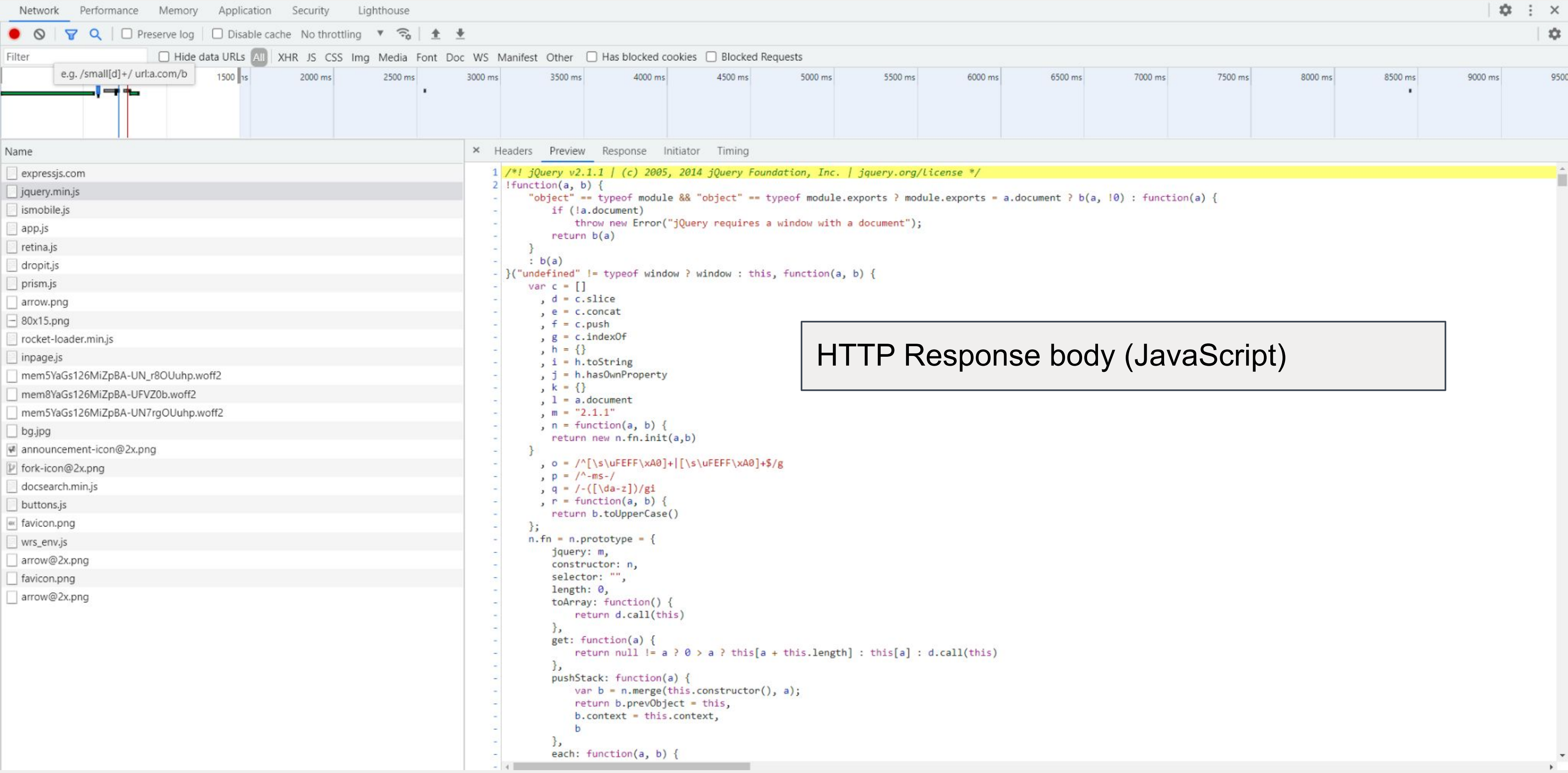
```html
<!DOCTYPE html>

<html lang="en">
<head>
<title>Express - Node.js web application framework</title>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-sca
<meta name="og:image" content="https://expressjs.com/images/ex
<link rel="icon" type="image/png" href="/images/favicon.png" /
<link rel="stylesheet" href="/css/style.css?_=1625153998">
<link rel="stylesheet" href="/css/dropit.css">
<link rel="stylesheet" href="/css/prism.css">
<link rel="stylesheet" href="/css/font-awesome.min.css">
<link rel="stylesheet" href="//fonts.googleapis.com/css?family=Open+Sans:300,400,600,700&amp;amp;subset=latin,latin-ext">
<link rel="stylesheet" href="/css/en.css">
<link rel="stylesheet" href="/css/nodeinteractive.css">
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<script data-cfasync="false" src="//ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js"></script>
<script data-cfasync="false" src="/js/ismobile.js"></script>
<script data-cfasync="false" src="/js/app.js"></script>
<script data-cfasync="false" src="/js/retina.js"></script>
<script data-cfasync="false" src="/js/dropit.js"></script>
<script data-cfasync="false" src="/js/prism.js"></script>
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/docsearch.js@2/dist/cdn/docsearch.min.css" />
</head>
<body class="en-doc">
<section id="home-content">
<header>
<div id="blm-banner">
Black Lives Matter. <br />
<a id="blm-donate" href="https://support.eji.org/give/153413/#!/donation/checkout">Support the Equal Justice Initiative</a>.
</div>
<div id="mobile-menu">
<div id="nav-button" class="fa fa-bars fa-2x button"></div>
</div>
<section id="logo"><a href="/" class="express">Express</a>
</section>
<div id="navbar">
<input id="q" placeholder="🔎 search" />
<ul id="navmenu">
<li><a href="/" id="home-menu" class="active">Home</a></li>
<li>
<ul id="getting-started-menu" class="menu">
<li><a href="/en/starter/installing.html">Getting started</a>
<ul>
<li>
```

HTTP Response body (HTML)



Name
expressjs.com
jquery.min.js
ismobile.js
app.js
retina.js
dropit.js
prism.js
arrow.png
80x15.png
rocket-loader.min.js
inpage.js
mem5YaGs126MiZpBA-UN_r8OUuhp.woff2
mem8YaGs126MiZpBA-UFVZ0b.woff2
mem5YaGs126MiZpBA-UN7rgOUuhp.woff2
bg.jpg
announcement-icon@2x.png
fork-icon@2x.png
docsearch.min.js
buttons.js
favicon.png
wrs_env.js
arrow@2x.png
favicon.png
arrow@2x.png

HTTP Response body (Image)



```
/*! jQuery v2.1.1 | (c) 2005, 2014 jQuery Foundation, Inc. | jquery.org/license */
!function(a, b) {
    "object" == typeof module && "object" == typeof module.exports ? module.exports = a.document ? b(a, !0) : function(a) {
        if (!a.document)
            throw new Error("jQuery requires a window with a document");
        return b(a)
    }
    : b(a)
}("undefined" != typeof window ? window : this, function(a, b) {
    var c = []
      , d = c.slice
      , e = c.concat
      , f = c.push
      , g = c.indexOf
      , h = {}
      , i = h.toString
      , j = h.hasOwnProperty
      , k = {}
      , l = a.document
      , m = "2.1.1"
      , n = function(a, b) {
        return new n.fn.init(a,b)
    }
      , o = /^[\s\uFEFF\xA0]+|[\s\uFEFF\xA0]+$/g
      , p = /^-ms-/
      , q = /-([\da-z])/gi
      , r = function(a, b) {
        return b.toUpperCase()
    };
    n.fn = n.prototype = {
        jquery: m,
        constructor: n,
        selector: "",
        length: 0,
        toArray: function() {
            return d.call(this)
        },
        get: function(a) {
            return null != a ? 0 > a ? this[a + this.length] : this[a] : d.call(this)
        },
        pushStack: function(a) {
            var b = n.merge(this.constructor(), a);
            return b.prevObject = this,
            b.context = this.context,
            b
        },
        each: function(a, b) {
```

HTTP Response body (JavaScript)