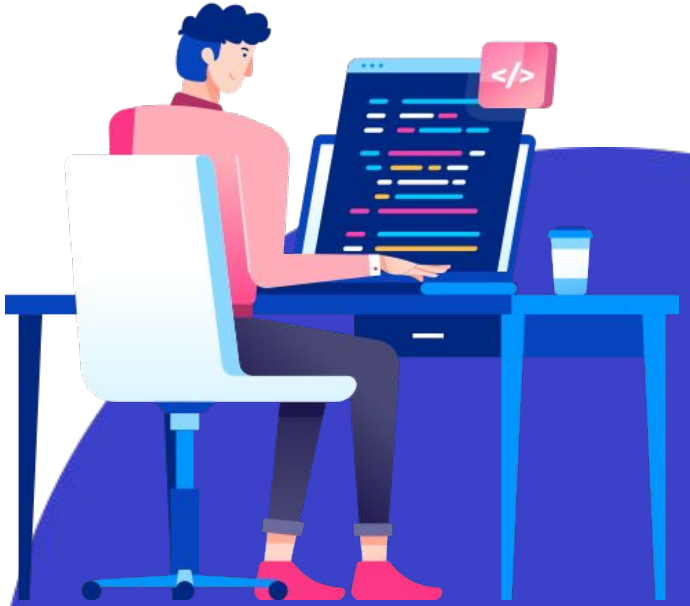
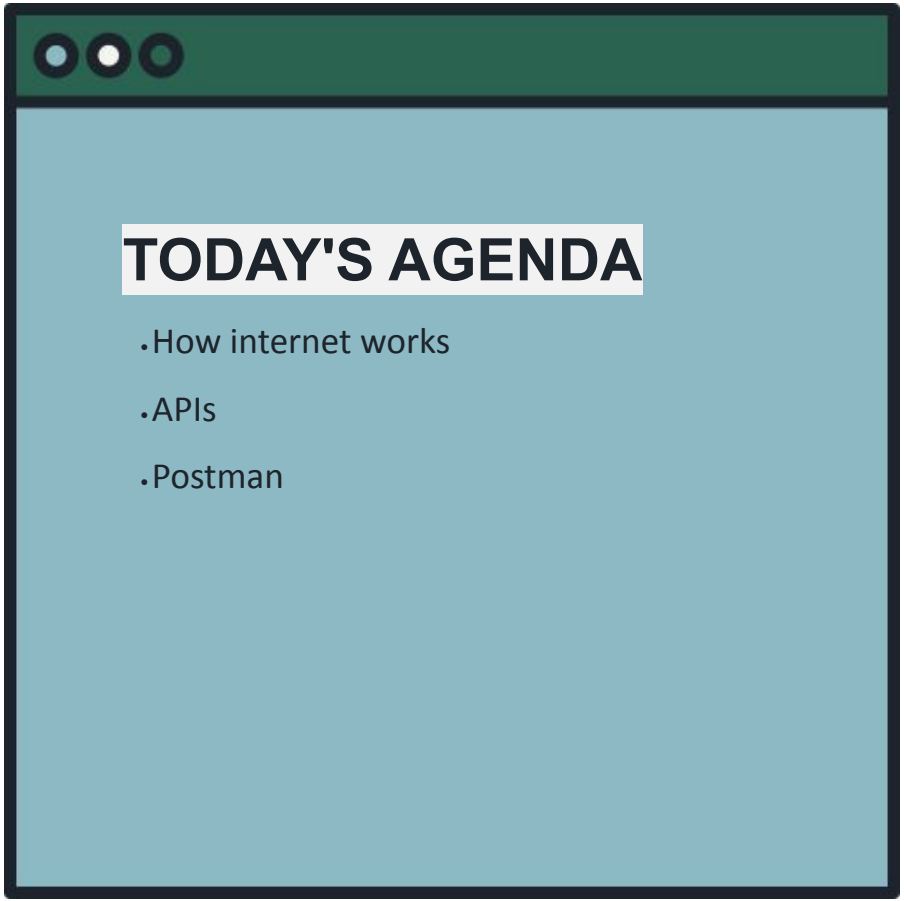
Programming Fundamentals

APIs

DIPLOMA IN FULL-STACK DEVELOPMENT

Certificate in **Computing Fundamentals**

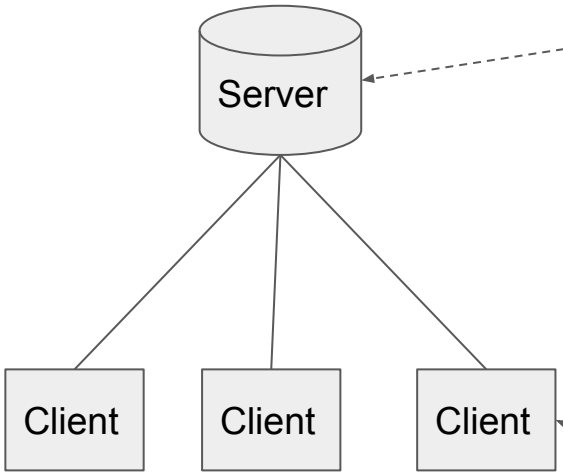




How the INTERNET WORKS

http protocol

Some Terminology



A website (html, js. css) etc. is hosted on a **server**

A software that connects to a server is known as a client

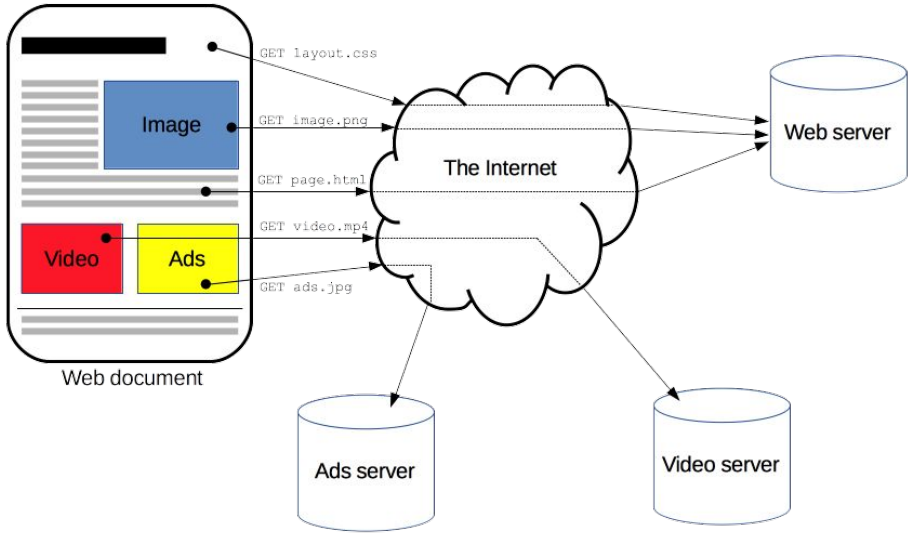
What is a Server

A server can be…

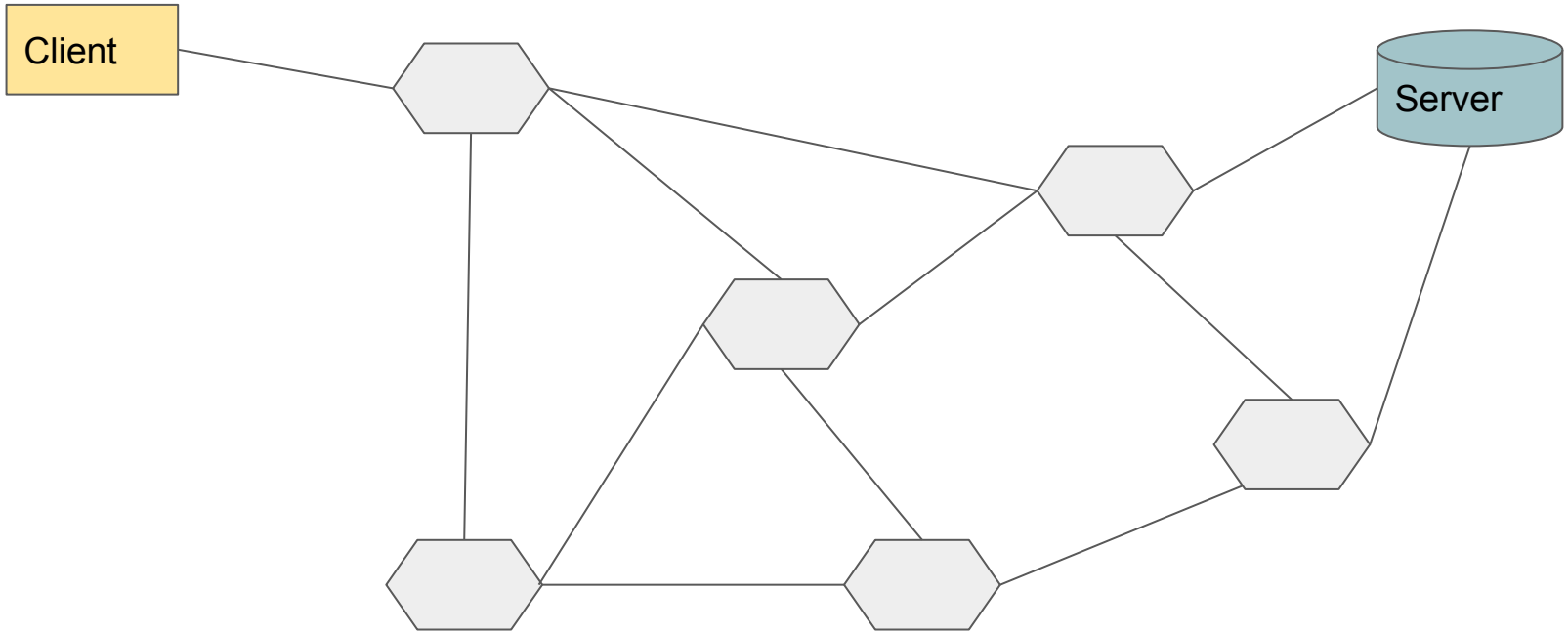
- A machine that is specially built to host a website, with plenty of RAM, high-speed HDD and etc.
- A software that allows other clients to connect

HTTP Protocol

In the context of networking, a **protocol** refers to how a client communicates with a server. It establishes the rules for the **format** of the resources that are being sent and received.



A resource here refers to data, such as a css file, html file, or image etc.



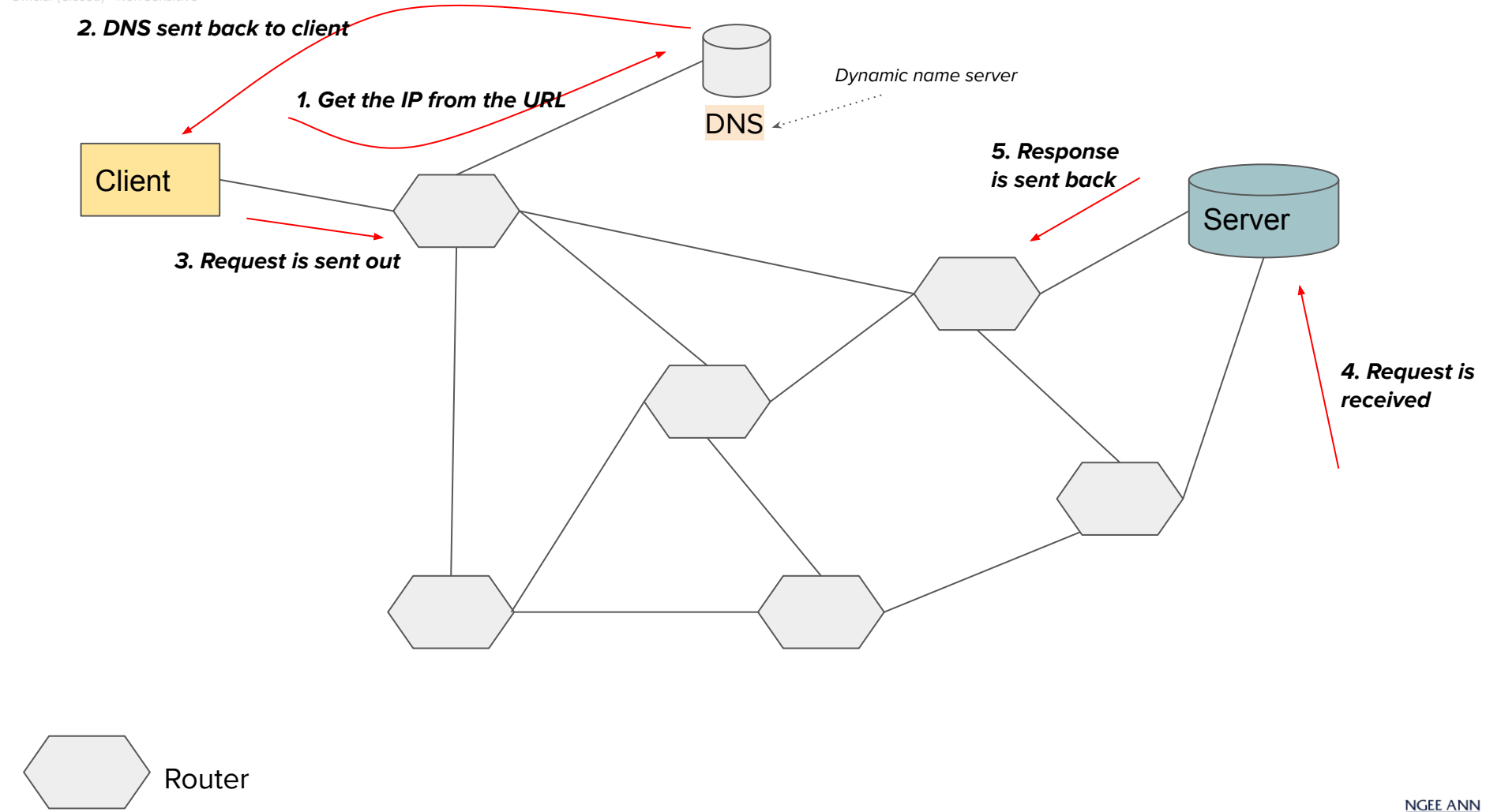
Router

IP Address & Web URLs

Each machine on the internet is given an **IP address.** We need to know the IP address of a machine access it.

However we usually access websites via **URLs.**

How does **URLs** get translated into **IP addresses?**



Some HTTP Properties

1. The data flow is different each time
   a. I.e, there is no **persistent connection**
2. It is **stateless**
   a. State refers to *data* stored by the data to **remember** some information about the client
   b. No state is maintained between client and server once the request is received and response is sent back.

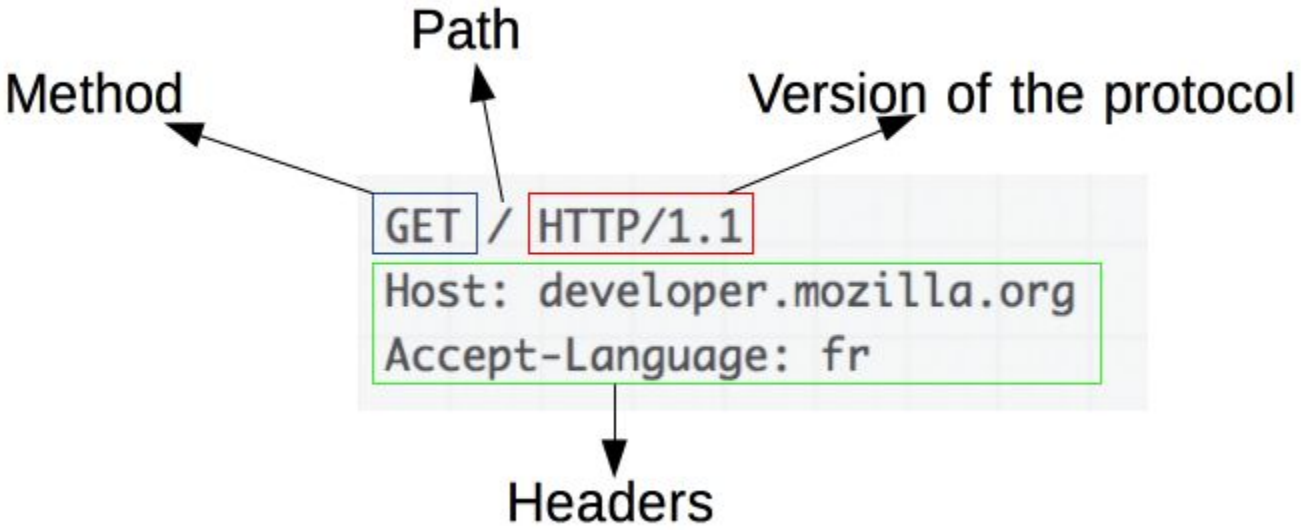
Request & Response

Request & Response

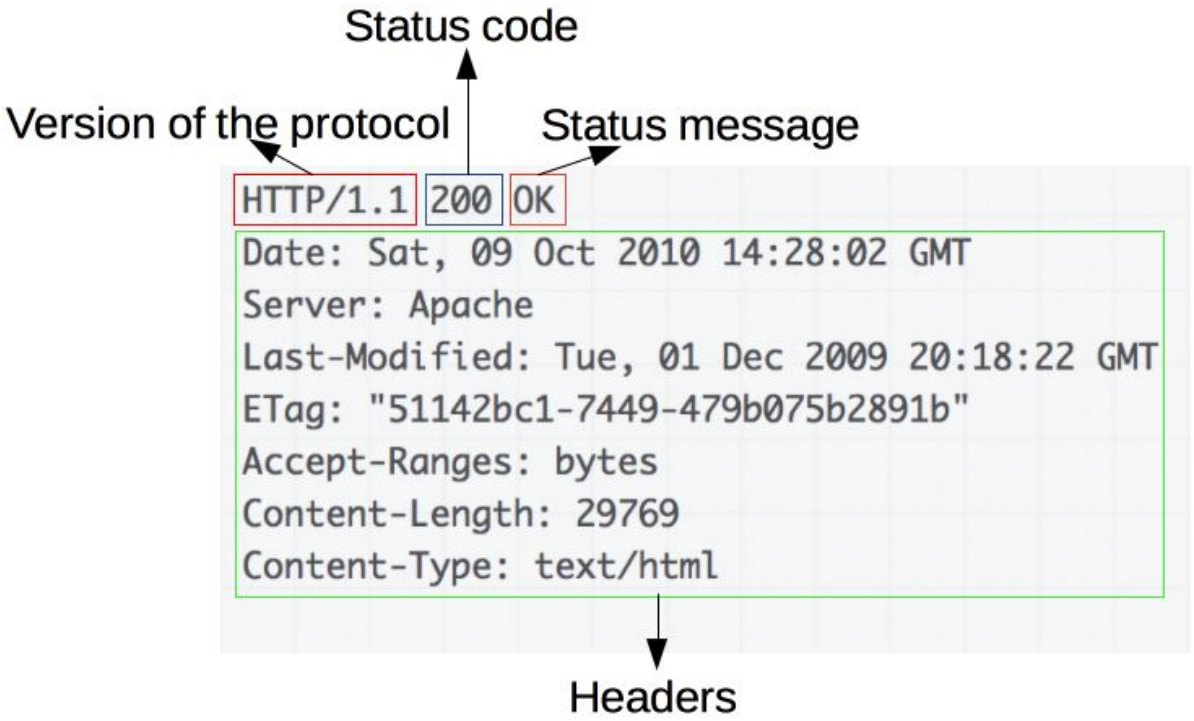
Request ⇒ What the client sends out

Response => What the client receives from the server (i.e, what does the server sends back to the client).

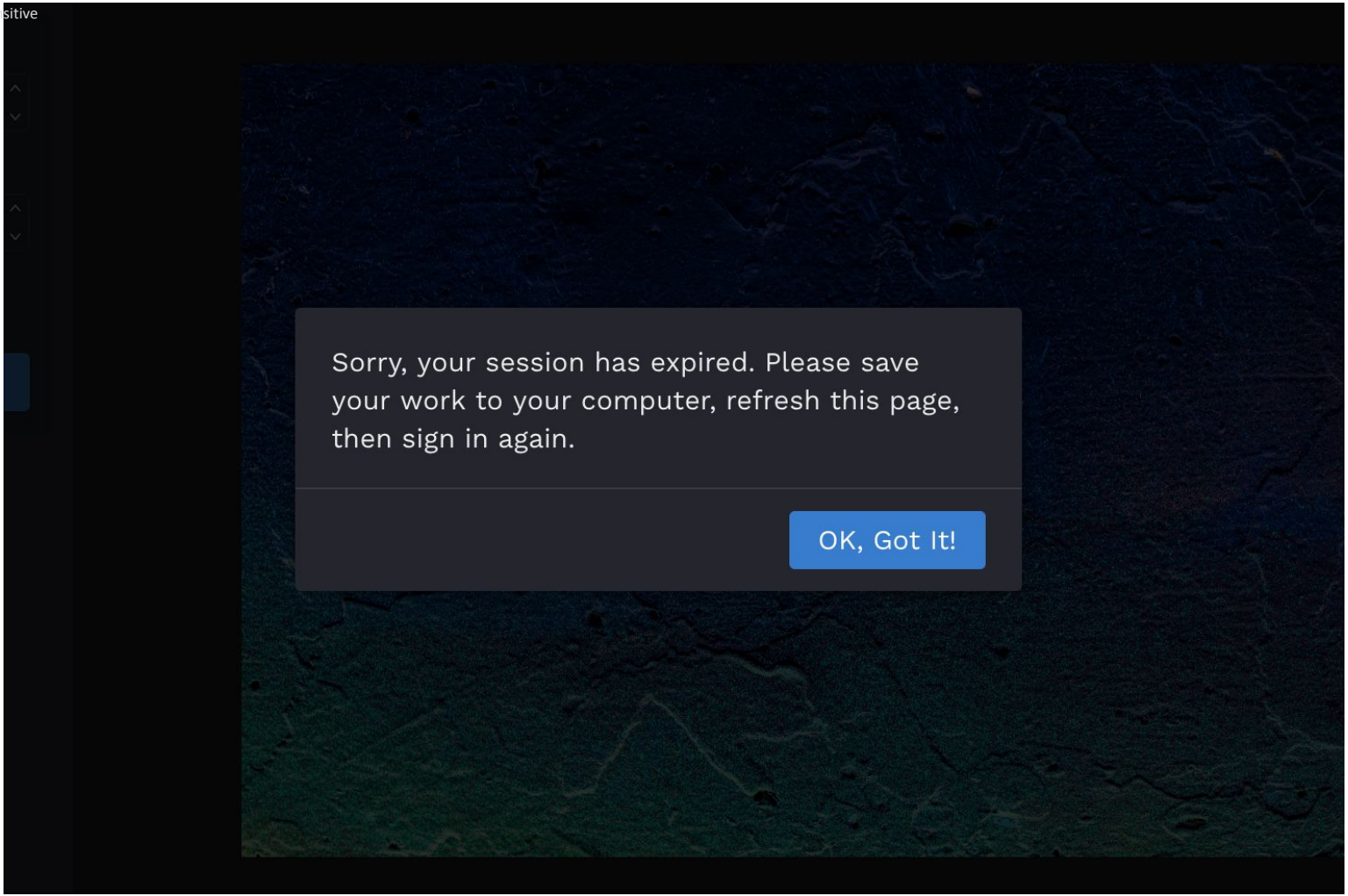
Sample Request



Sample Response



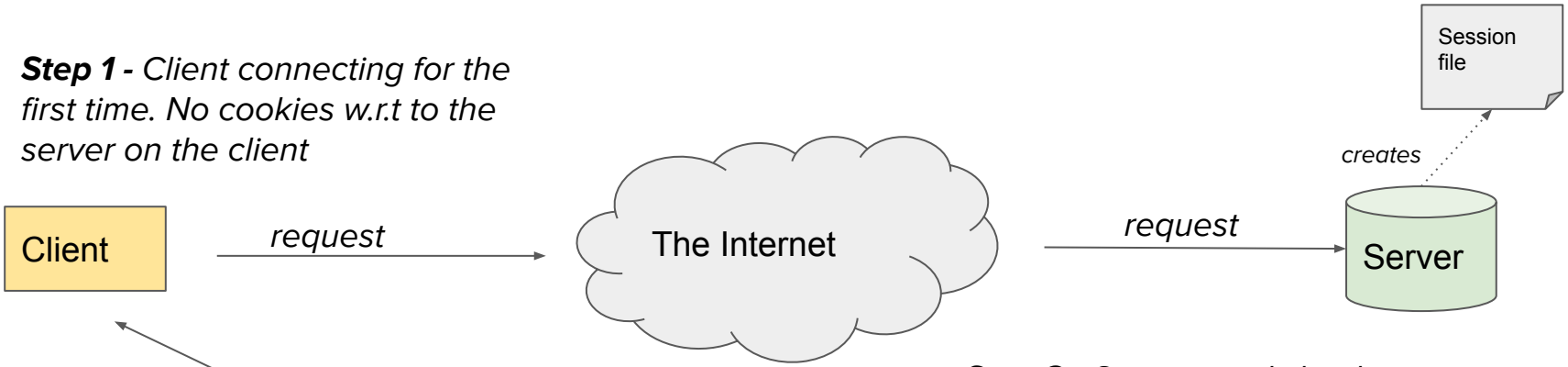
Sessions



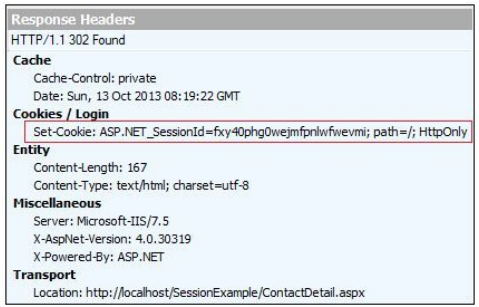
Session & Cookies

- Sessions represented **state** stored by the server.
- Every client that connects to the server has a **session.**
  - The session is stored as a file on the server.
- Corresponding to each session is a **cookie file**
  - A cookie allows the client to remember about the server

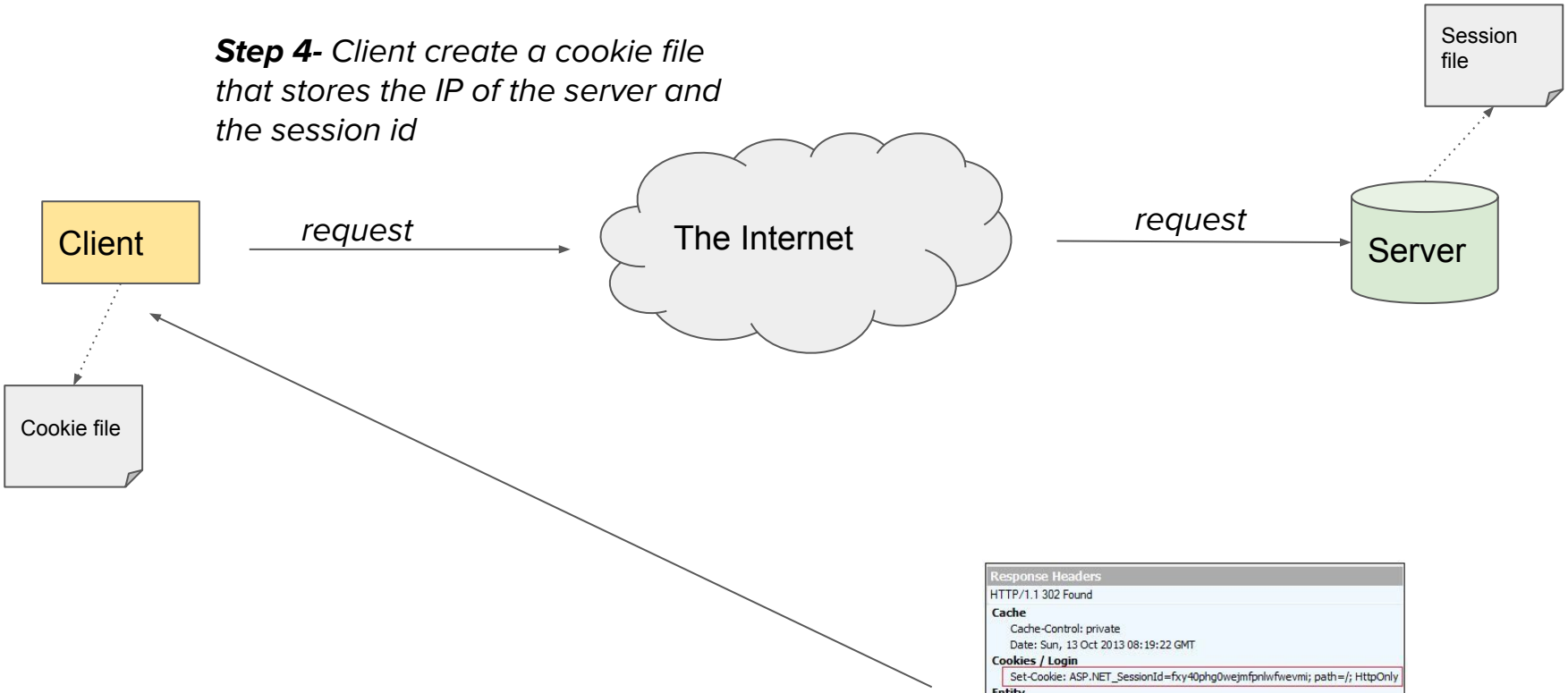
Step 1 - *Client connecting for the first time. No cookies w.r.t to the server on the client*



Step 2 - *Server sends back response.* ***Because client has never connected, it creates a session file and sends back an session_id and a session file***



Step 4- *Client create a cookie file that stores the IP of the server and the session id*



Response Headers

HTTP/1.1 302 Found

Cache
- Cache-Control: private
- Date: Sun, 13 Oct 2013 08:19:22 GMT

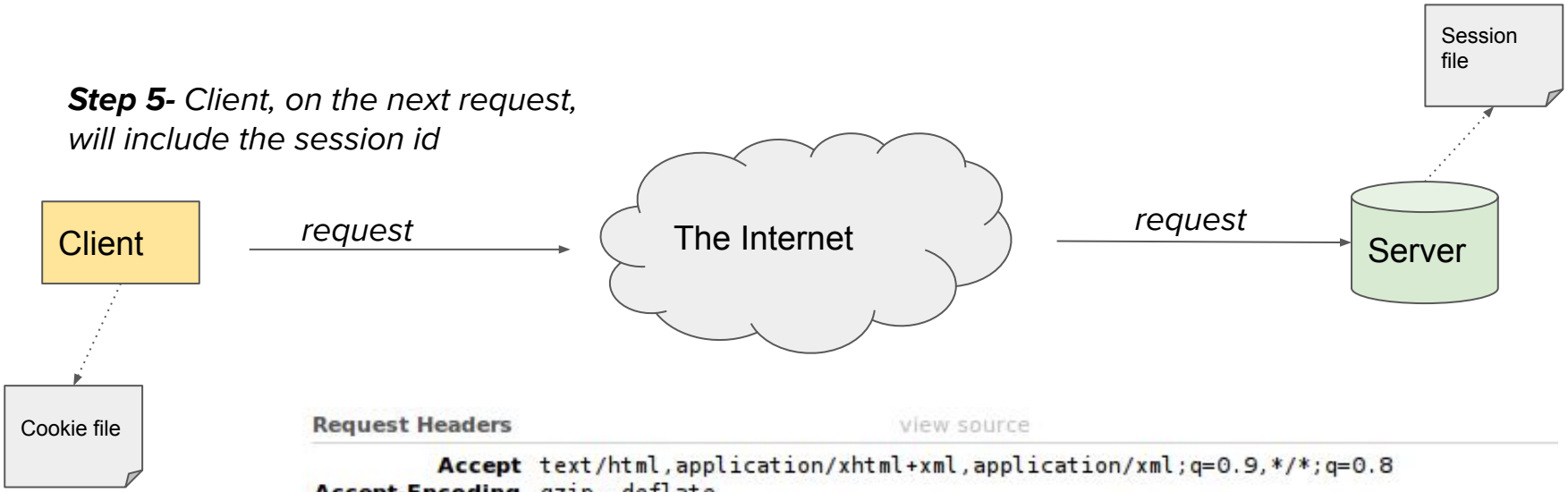
Cookies / Login
- Set-Cookie: ASP.NET_SessionId=fxy40phg0wejmfpnlwfwevmi; path=/; HttpOnly

Entity
- Content-Length: 167
- Content-Type: text/html; charset=utf-8

Miscellaneous
- Server: Microsoft-IIS/7.5
- X-AspNet-Version: 4.0.30319
- X-Powered-By: ASP.NET

Transport
- Location: http://localhost/SessionExample/ContactDetail.aspx

Step 5- *Client, on the next request, will include the session id*



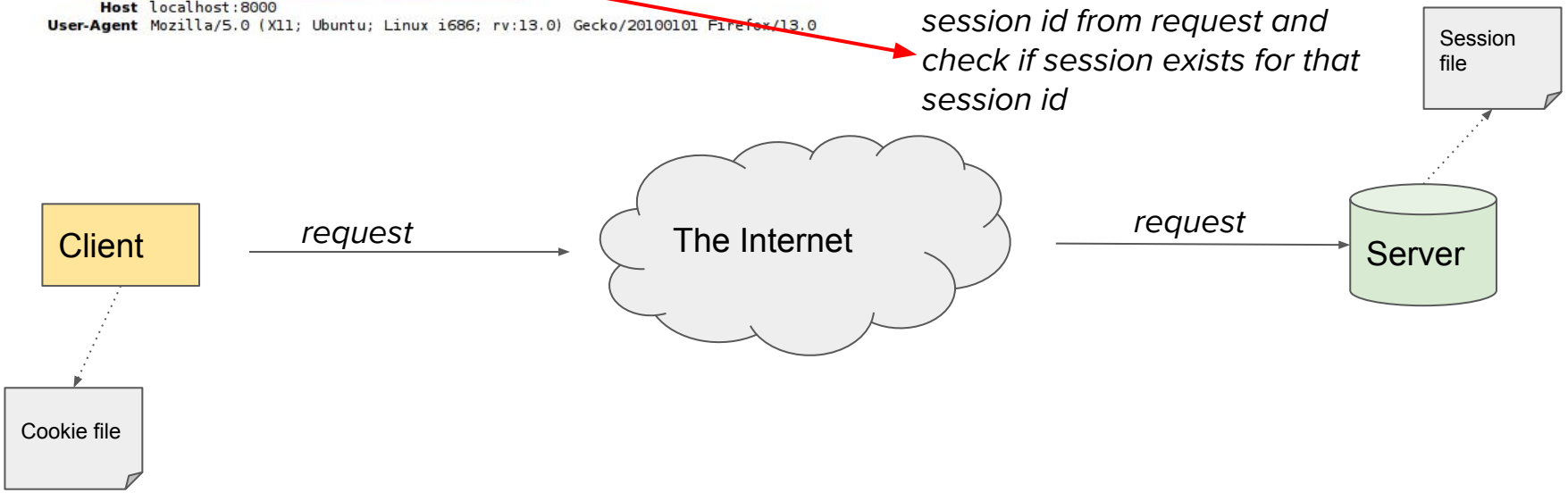
Request Headers view source

Accept	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding	gzip, deflate
Accept-Language	en-us,en;q=0.5
Cache-Control	max-age=0
Connection	keep-alive
Cookie	sessionid=4d03b5ac14932f10ec91293344c9433a
Host	localhost:8000
User-Agent	Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:13.0) Gecko/20100101 Firefox/13.0

Request Headers view source

Accept	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding	gzip, deflate
Accept-Language	en-us,en;q=0.5
Cache-Control	max-age=0
Connection	keep-alive
Cookie	sessionid=4d03b5ac14932f10ec91293344c9433a
Host	localhost:8000
User-Agent	Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:13.0) Gecko/20100101 Firefox/13.0

Step 6- *Server will take the session id from request and check if session exists for that session id*



HTTP Status Codes

Status Code

Whenever a server sends back a response, it also **associates a *status code*** with it.

The status code indicates whether the request has been successful or not.

Response Code

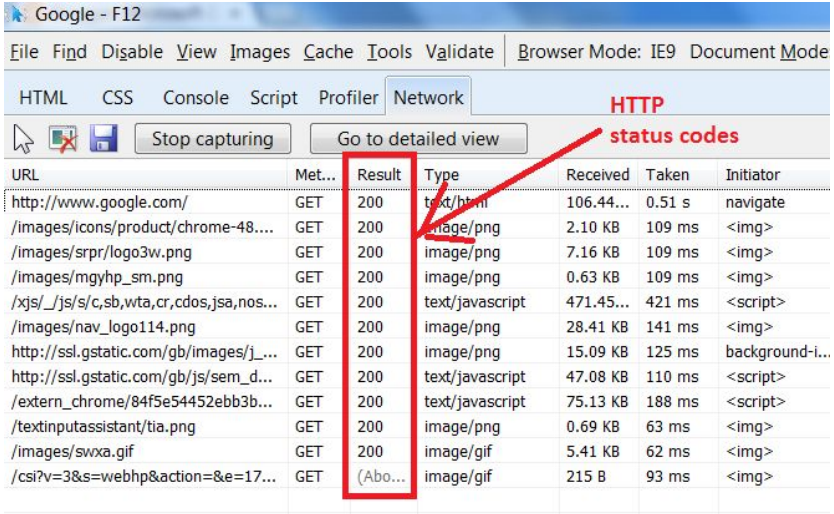
100s - request is on-going

200s - OK

300s - redirects (trying a different URL)

400s - not found or authorization errors

500s - internal server errors (something went wrong at the server side)



https://developer.mozilla.org/en-US/docs/Web/HTTP/Status

APIs and AJAX

More stuff...

Asynchronous JavaScript and XML

! NOT a framework
! NOT a library
! NOT a technology

AJAX

AJAX is a technique for loading data into part of a page without having to refresh the entire page. The data is often sent in a format called JavaScript Object Notation (or JSON)

The ability to load new content into part of a page improves the user experience because the user does not have to wait for an entire page to load if only a portion of the page is being updated.

https://replit.com/@immalcolm/ajax-examples

AJAX IN ACTION

Live search (or autocomplete) commonly uses Ajax. Eg. Google search. When you type into the search bar on the home page, sometimes you will see results coming up before you have finished typing

Websites with user-generated content (such as Twitter and Flickr) may allow you to display your information (such as your latest tweets or photographs) This involves collecting data from their servers or tapping on APIs

https://replit.com/@immalcolm/ajax-examples

Image Source: https://developer.twitter.com/en/docs

WHY USE AJAX

USING AJAX WHILE PAGES ARE LOADING

When a browser comes across a <script> tag, it will typically stop processing the rest of the page until it has loaded and processed that script. This is known as a **synchronous processing model.**

With AJAX, the browser can **request** some data from a server and once that data has been **requested**, continue to load the rest of the page and process the user's interactions. This is known as asynchronous (or non-blocking) processing model.

https://replit.com/@immalcolm/ajax-examples

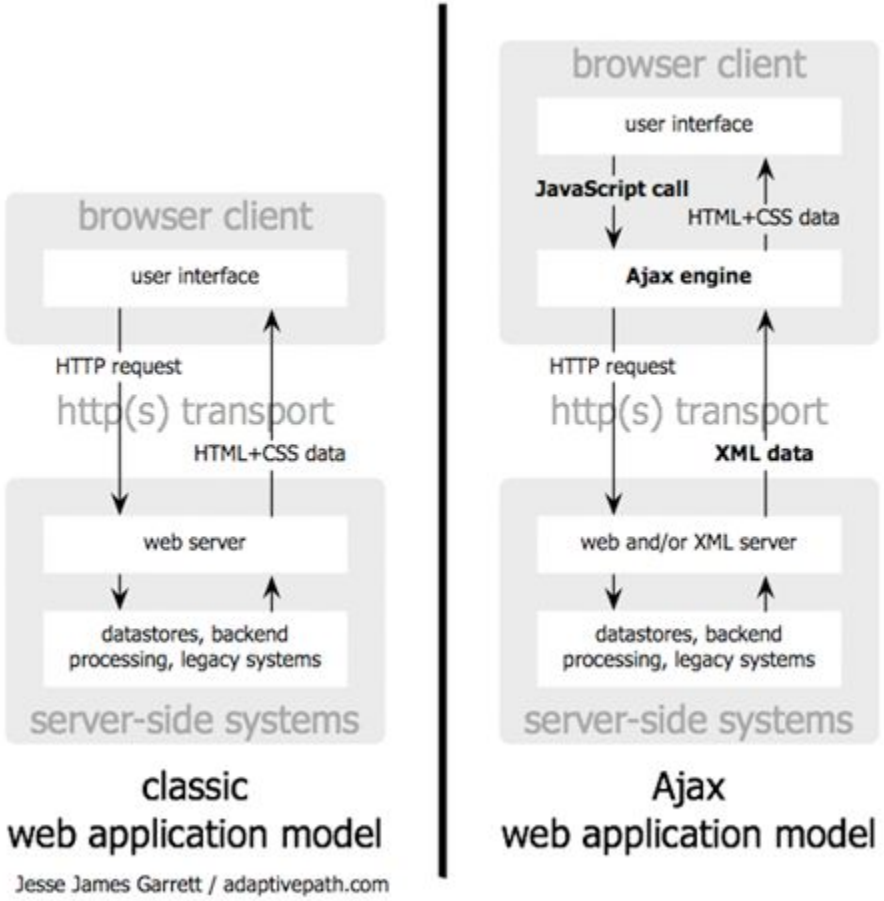
WHY USE AJAX

USING AJAX WHEN PAGES HAVE LOADED

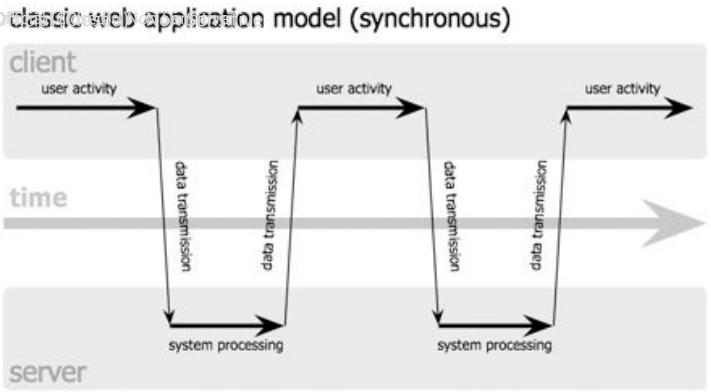
Once a page has loaded, if you want to update what the user sees in the browser, typically you would refresh the entire page. This means the user has to wait for a whole new page to download and be rendered by the browser.

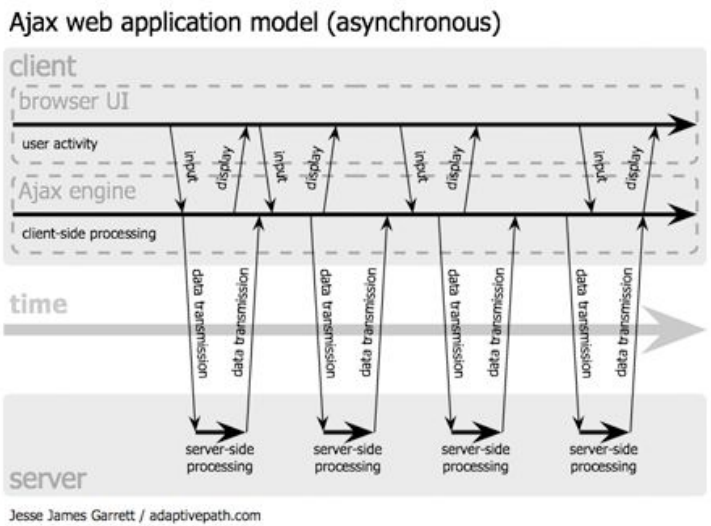
With AJAX, if you only need to update a part of the page, you can just update the content of one element. This is done by intercepting an event (user clicks on link or submits a form) and requesting new content from the server via an asynchronous request.

https://replit.com/@immalcolm/ajax-examples



Reading Source & Image Credits: http://adaptivepath.org/ideas/ajax-new-approach-web-applications/





An Ajax application eliminates the start-stop-start-stop nature of interaction on the Web by introducing an intermediary — an Ajax engine — between the user and the server.

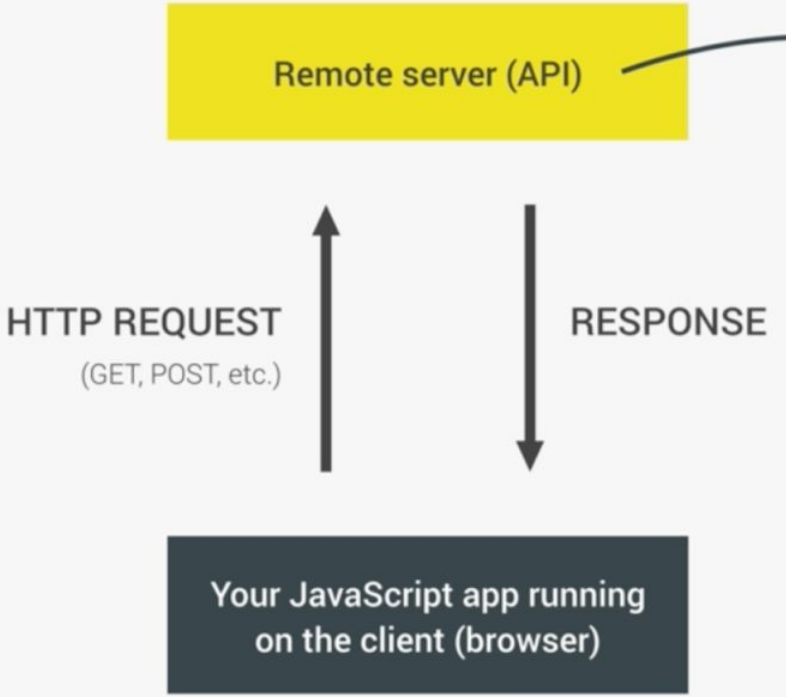
Instead of loading a webpage, at the start of the session, the browser loads an Ajax engine — written in JavaScript and usually tucked away in a hidden frame. This engine is responsible for both rendering the interface the user sees and communicating with the server on the user's behalf.

Reading Source & Image Credits: http://adaptivepath.org/ideas/ajax-new-approach-web-applications/

AJAX & APIs

ASYNCHRONOUS **J**AVASCRIPT **A**ND **X**ML

APPLICATION **P**ROGRAMMING **I**NTERFACE



- Your own API, for data coming from your own server
- 3rd-party APIs:
  - Google Maps
  - Embed Youtube videos
  - Weather data
  - Movies data
  - Send email or SMS
  - Thousands of possibilities...

Asynchronous JavaScript and XML

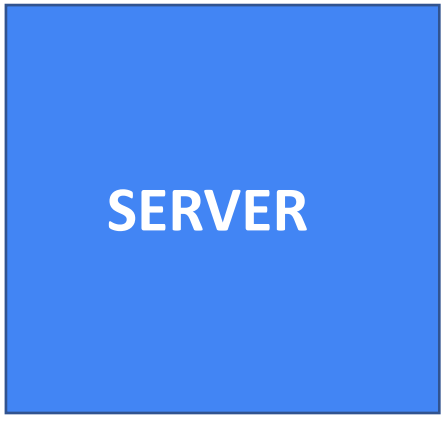
AJAX SIMPLIFIED

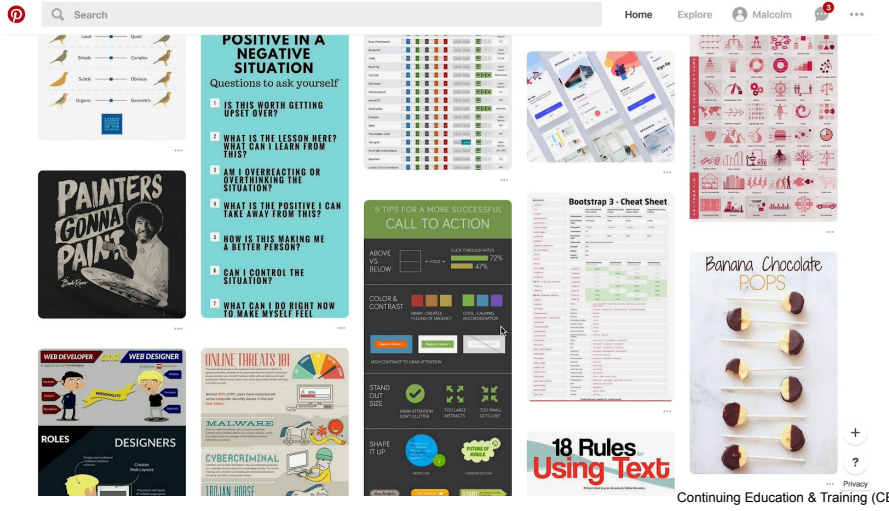
With AJAX, websites can send and request data from a server in the background without disturbing the current page

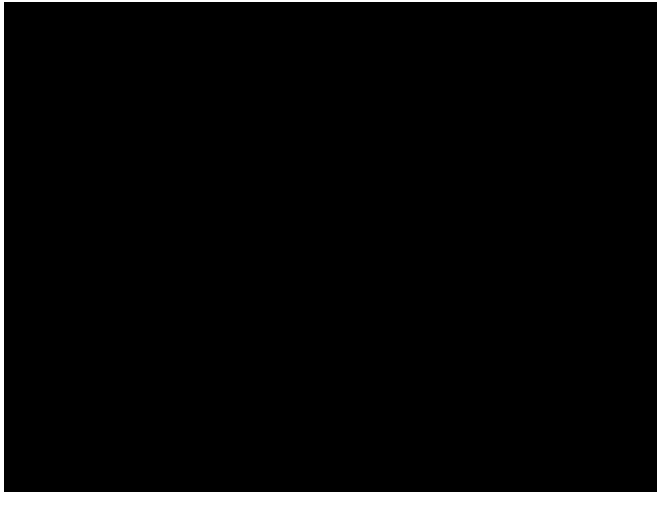
Today's Single Page Application (SPA)
E.g; twitter, facebook, gmail, pinterest, instagram

AJAX EXAMPLE

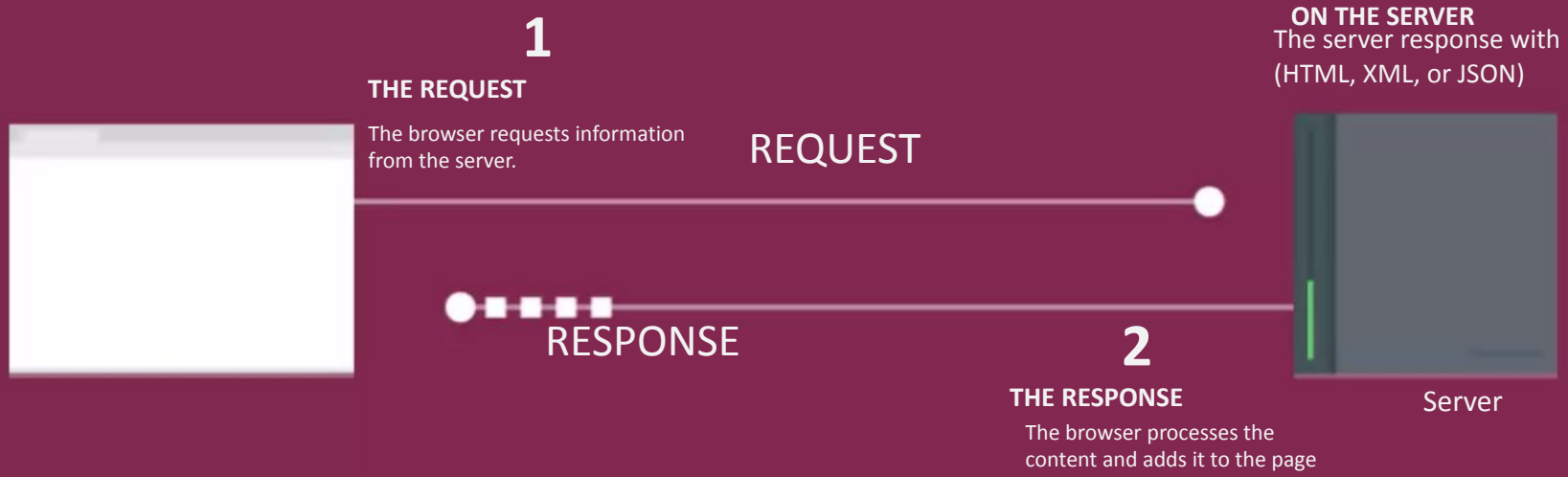
1. The user scrolled to the bottom
2. Quick, send a request to the server and get more content
3. Now I'll just append the new data to the bottom of the page.







HOW AJAX WORKS



AJAX lets you build webpages that ask for information from a web server. The web server returns data to the web browser, and JavaScript processes that data to selectively change parts of the web page.

The amount of data the server returns is usually much less than that sent when you ask for a full webpage.

The process of asking a server for information is technically called making a **request** of the server.

And when the server sends back its answer, that's called a **response**.

https://replit.com/@immalcolm/ajax-examples

HOW AJAX WORKS

1. Create an XMLHTTP Request → This step tells the web browser to get ready. You want to send an AJAX request and the web browser needs to create *a*n object that has all the methods you'll need to send and receive data
2. Create a callback function → This is the programming you want to run when the server returns its response. The callback is where you process the returned data and update the HTML on the page.
3. Open a request → Browser will send the request using POST or GET
4. Send the request → Send request to the server

https://replit.com/@immalcolm/ajax-examples

DATA FORMATS

HTML

Simplest way to get data from a page

BENEFITS

- Easy to write, request and display
- Data sent from the server goes straight into the page. There's no need for the browser to process it

DRAWBACKS

- Server must produce the HTML in a format that is ready for use on your page
- Not suited for use in applications other than web browsers. Poor data portability
- Request must come from the same domain

XML

Similar to HTML but tag name are different.

BENEFITS

- Flexible data form and can represent complex data structures
- Works well with different platforms and applications
- Processed using the same DOM methods as HTML

DRAWBACKS

- Considered as verbose language due to the tags adding lots of extra characters
- Request must come from the same domain as the rest of the page
- Requires a lot of code to process the result

JSON

JavaScript Object Notation (JSON) similar syntax to literal object

BENEFITS

- Can be called from any domain
- More concise than HTML /XML
- Commonly used

DRAWBACKS

- Syntax is not forgiving
- May contain malicious content, always use from trusted sources

https://replit.com/@immalcolm/ajax-examples

XML

```xml
<?xml version="1.0" encoding="utf-8" ?>
<events>
  <event>
    <location>Woodlands</location>
    <date>Nov 10</date>
    <map>img/woodlands.png</map>
  </event>
  <event>
    <location>Clementi</location>
    <date>Nov 8</date>
    <map>img/clementi.png</map>
  </event>
  <event>
    <location>Bedok</location>
    <date>Nov 1</date>
    <map>img/bedok.png</map>
  </event>
</events>
```

JSON

```json
{
  "events": [
    {
      "location": "Woodlands",
      "date": "Nov 10",
      "map": "img/woodlands.png"
    },
    {
      "location": "Clementi",
      "date": "Nov 8",
      "map": "img/clementi.png"
    },
    {
      "location": "Bedok",
      "date": "Nov 1",
      "map": "img/bedok.png"
    },

  ]
}
```

https://replit.com/@immalcolm/ajax-examples

SAMPLE JSON FROM REDDIT

Sample Data: https://www.reddit.com/.json


```
{"kind": "Listing", "data": {"modhash": "", "dist": 25, "children": [{"kind": "t3", "data": {"is_crosspostable": false, "subreddit_id": "t5_2viuz",
"approved_at_utc": null, "wls": 6, "mod_reason_by": null, "banned_by": null, "num_reports": null, "removal_reason": null, "thumbnail_width": 140, "subredd
"combinedgifs", "selftext_html": null, "author_flair_template_id": null, "selftext": "", "likes": null, "suggested_sort": null, "user_reports": [],
"secure_media": null, "is_reddit_media_domain": false, "saved": false, "id": "8j3wj0", "banned_at_utc": null, "mod_reason_title": null, "gilded": 0, "arch
false, "clicked": false, "no_follow": false, "author": "Baked_anne_frink", "num_crossposts": 1, "link_flair_text": null, "can_mod_post": false, "send_repl
false, "pinned": false, "score": 15640, "approved_by": null, "over_18": false, "report_reasons": null, "domain": "i.imgur.com", "hidden": false, "preview"
{"images": [{"source": {"url": "https://i.redditmedia.com/nheLB-f5VDfqwIAg_irAp7rtZqjtqF5CAhcjzB235Zk.gif?fm=jpg&amp;s=f3369fbdd5a10a9b0e223ff32125ab5f",
728, "height": 408}, "resolutions": [{"url": "https://i.redditmedia.com/nheLB-f5VDfqwIAg_irAp7rtZqjtqF5CAhcjzB235Zk.gif?
fit=crop&amp;crop=faces%2Centropy&amp;arh=2&amp;w=108&amp;fm=jpg&amp;s=2cdf73bf73cf97a088458dcb2475e46c", "width": 108, "height": 60}, {"url":
"https://i.redditmedia.com/nheLB-f5VDfqwIAg_irAp7rtZqjtqF5CAhcjzB235Zk.gif?
fit=crop&amp;crop=faces%2Centropy&amp;arh=2&amp;w=216&amp;fm=jpg&amp;s=30d1242a52040e7c2a27b4ae4785553c", "width": 216, "height": 121}, {"url":
"https://i.redditmedia.com/nheLB-f5VDfqwIAg_irAp7rtZqjtqF5CAhcjzB235Zk.gif?
fit=crop&amp;crop=faces%2Centropy&amp;arh=2&amp;w=320&amp;fm=jpg&amp;s=7d00da2722e1a0b6a56c06fadf55726a", "width": 320, "height": 179}, {"url":
"https://i.redditmedia.com/nheLB-f5VDfqwIAg_irAp7rtZqjtqF5CAhcjzB235Zk.gif?
fit=crop&amp;crop=faces%2Centropy&amp;arh=2&amp;w=640&amp;fm=jpg&amp;s=98908f798448ce240d9aeb59e920ffdd", "width": 640, "height": 358}], "variants": {"gif
{"source": {"url": "https://g.redditmedia.com/nheLB-f5VDfqwIAg_irAp7rtZqjtqF5CAhcjzB235Zk.gif?s=310b7b8d7a1274e6f69524a1f4873c78", "width": 728, "height":
"resolutions": [{"url": "https://g.redditmedia.com/nheLB-f5VDfqwIAg_irAp7rtZqjtqF5CAhcjzB235Zk.gif?
fit=crop&amp;crop=faces%2Centropy&amp;arh=2&amp;w=108&amp;s=80bf2dfdbdf4d7a0086ed4391d50add7", "width": 108, "height": 60}, {"url":
"https://g.redditmedia.com/nheLB-f5VDfqwIAg_irAp7rtZqjtqF5CAhcjzB235Zk.gif?
fit=crop&amp;crop=faces%2Centropy&amp;arh=2&amp;w=216&amp;s=a1def6fc3e03a19d975ee5d83e9e8714", "width": 216, "height": 121}, {"url":
"https://g.redditmedia.com/nheLB-f5VDfqwIAg_irAp7rtZqjtqF5CAhcjzB235Zk.gif?
fit=crop&amp;crop=faces%2Centropy&amp;arh=2&amp;w=320&amp;s=c86c874b90c1197239e0af2f4064f199", "width": 320, "height": 179}, {"url":
"https://g.redditmedia.com/nheLB-f5VDfqwIAg_irAp7rtZqjtqF5CAhcjzB235Zk.gif?
fit=crop&amp;crop=faces%2Centropy&amp;arh=2&amp;w=640&amp;s=166cdfa5fb98ffe7b81531d332814871", "width": 640, "height": 358}]}, "mp4": {"source": {"url":
"https://g.redditmedia.com/nheLB-f5VDfqwIAg_irAp7rtZqjtqF5CAhcjzB235Zk.gif?fm=mp4&amp;mp4-fragmented=false&amp;s=ea0d0e18aae2c03762086e862efe9445", "width
"height": 408}, "resolutions": [{"url": "https://g.redditmedia.com/nheLB-f5VDfqwIAg_irAp7rtZqjtqF5CAhcjzB235Zk.gif?
fit=crop&amp;crop=faces%2Centropy&amp;arh=2&amp;w=108&amp;fm=mp4&amp;mp4-fragmented=false&amp;s=23f1a925dec5e1b71b4cb81a2f980125", "width": 108, "height":
{"url": "https://g.redditmedia.com/nheLB-f5VDfqwIAg_irAp7rtZqjtqF5CAhcjzB235Zk.gif?fit=crop&amp;crop=faces%2Centropy&amp;arh=2&amp;w=216&amp;fm=mp4&amp;mp
fragmented=false&amp;s=317690cc99f7d6dcd5390bff8bc4a4ae", "width": 216, "height": 121}, {"url": "https://g.redditmedia.com/nheLB-
f5VDfqwIAg_irAp7rtZqjtqF5CAhcjzB235Zk.gif?fit=crop&amp;crop=faces%2Centropy&amp;arh=2&amp;w=320&amp;fm=mp4&amp;mp4-
fragmented=false&amp;s=0d87ede34ae0a1d1cc8e66d49aeb0358", "width": 320, "height": 179}, {"url": "https://g.redditmedia.com/nheLB-
f5VDfqwIAg_irAp7rtZqjtqF5CAhcjzB235Zk.gif?fit=crop&amp;crop=faces%2Centropy&amp;arh=2&amp;w=640&amp;fm=mp4&amp;mp4-
fragmented=false&amp;s=c671b39bb8ae7b11973dc816df82e978", "width": 640, "height": 358}]}}, "id": "GuhVz6DiJm8bXook2P64-nSkAy7LcVpQ4a8VaT4kQHQ"}],
"reddit_video_preview": {"fallback_url": "https://v.redd.it/6781ju4romx01/DASH_1_2_M", "height": 360, "width": 640, "scrubber_media_url":
"https://v.redd.it/6781ju4romx01/DASH_600_K", "dash_url": "https://v.redd.it/6781ju4romx01/DASHPlaylist.mpd", "duration": 8, "hls_url":
"https://v.redd.it/6781ju4romx01/HLSPlaylist.m3u8", "is_gif": true, "transcoding_status": "completed"}, "enabled": true}, "pwls": 6, "thumbnail":
"https://b.thumbs.redditmedia.com/iFnANL88HEpml_E0C03wILDB8ZehbQlOf26S4_iFCfE.jpg", "edited": false, "link_flair_css_class": null, "author_flair_css_class
"contest_mode": false, "view_count": null, "locked": false, "downs": 0, "mod_reports": [], "subreddit_subscribers": 312594, "secure_media_embed": {},
"media_embed": {}, "post_hint": "link", "stickied": false, "visited": false, "can_gild": false, "thumbnail_height": 78, "name": "t3_8j3wj0", "spoiler": fa
"permalink": "/r/combinedgifs/comments/8j3wj0/an_amazing_dive/", "subreddit_type": "public", "parent_whitelist_status": "all_ads", "hide_score": false, "c
1526248289.0, "url": "https://i.imgur.com/79GNwZK.gifv", "author_flair_text": null, "quarantine": false, "title": "An amazing dive.", "created_utc": 15262
"previous_visits": [], "subreddit_name_prefixed": "r/combinedgifs", "ups": 15640, "num_comments": 185, "media": null, "is_self": false, "whitelist_status"
"all_ads", "mod_note": null, "is_video": false, "distinguished": null, "post_categories": null}}, {"kind": "t3", "data": {"is_crosspostable": false,
"subreddit_id": "t5_2ylxz", "approved_at_utc": null, "wls": 6, "mod_reason_by": null, "banned_by": null, "num_reports": null, "removal_reason": null,
"thumbnail_width": 140, "subreddit": "HighQualityGifs", "selftext_html": null, "author_flair_template_id": null, "selftext": "", "likes": null, "suggested
null, "user_reports": [], "secure_media": null, "is_reddit_media_domain": false, "saved": false, "id": "8j3em8", "banned_at_utc": null, "mod_reason_title"
"view_count": null, "archived": false, "clicked": false, "no_follow": false, "author": "rooster_86", "num_crossposts": 1, "link_flair_text": "Inglourious
Basterds", "can_mod_post": false, "send_replies": true, "pinned": false, "score": 25321, "approved_by": null, "over_18": false, "report_reasons": null, "d
"i.imgur.com", "hidden": false, "preview": {"images": [{"source": {"url": "https://i.redditmedia.com/_ssWa3DoPkTle8TovSDOW-fCvKtDjqEoftNCf6L7FnA.jpg?
s=d25a988f70c3276282c26aeb17e4e05a", "width": 918, "height": 350}, "resolutions": [{"url": "https://i.redditmedia.com/_ssWa3DoPkTle8TovSDOW-
fCvKtDjqEoftNCf6L7FnA.jpg?fit=crop&amp;crop=faces%2Centropy&amp;arh=2&amp;w=108&amp;s=ff9bbb4900b6fcb7e784e06035eaa60a", "width": 108, "height": 41}, {"ur
```

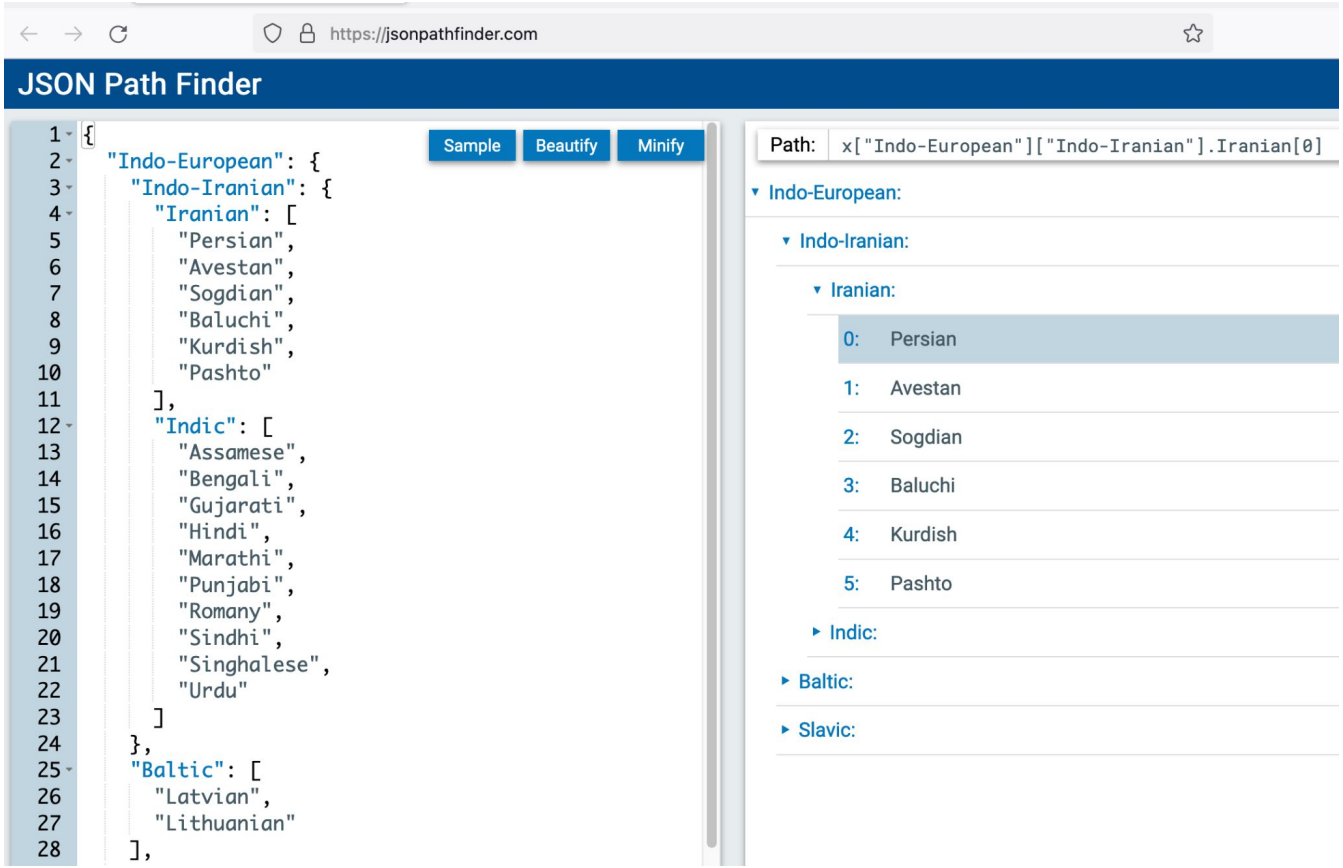


```
kind:                                   "Listing"
data:
  modhash:                              ""
  dist:                                 25
  children:
    0:
      kind:                             "t3"
      data:
        is_crosspostable:               false
        subreddit_id:                   "t5_2viuz"
        approved_at_utc:                null
        wls:                            6
        mod_reason_by:                  null
        banned_by:                      null
        num_reports:                    null
        removal_reason:                 null
        thumbnail_width:                140
        subreddit:                      "combinedgifs"
        selftext_html:                  null
        author_flair_template_id:       null
        selftext:                       ""
        likes:                          null
        suggested_sort:                 null
        user_reports:                   []
        secure_media:                   null
        is_reddit_media_domain:         false
        saved:                          false
        id:                             "8j3wj0"
        banned_at_utc:                  null
        mod_reason_title:               null
        gilded:                         0
        archived:                       false
        clicked:                        false
        no_follow:                      false
        author:                         "Baked_anne_frink"
        num_crossposts:                 0
        link_flair_text:                null
        can_mod_post:                   false
```

🕐 Understanding JSON & Pathing

https://jsonpathfinder.com/

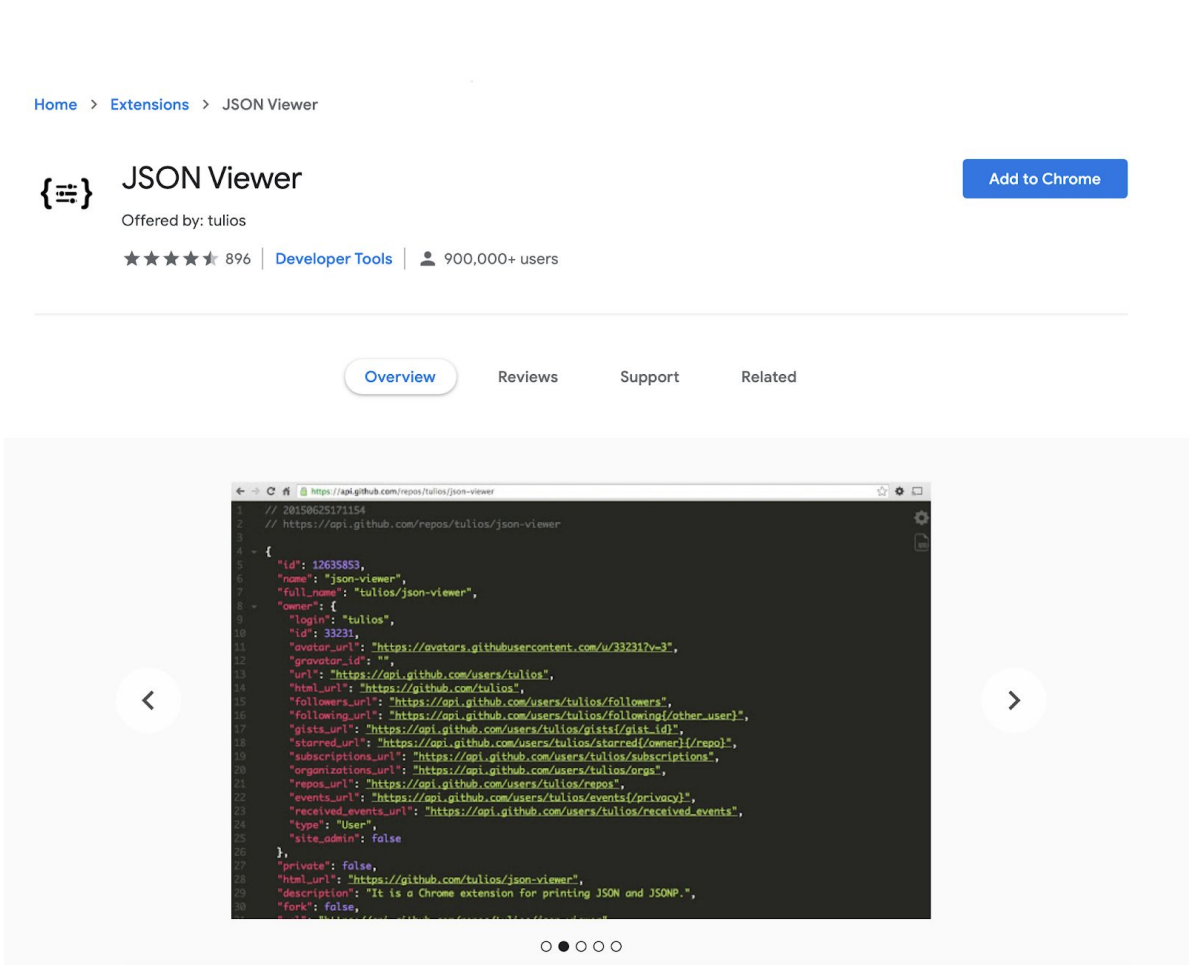
Getting stuck or confused with array and object mapping or finding the index? JSON Path Finder is a great utility to better visualise how pathing works in a JSON



JSON Toolkit

Chrome Extension JSON Viewer

http://jsonviewer.stack.hu/



Download JSON Viewer Extensions
https://chrome.google.com/webstore/detail/json-viewer/gbmdgpbipfallnflgajpaliibnhdgobh

Making Requests with JavaScript

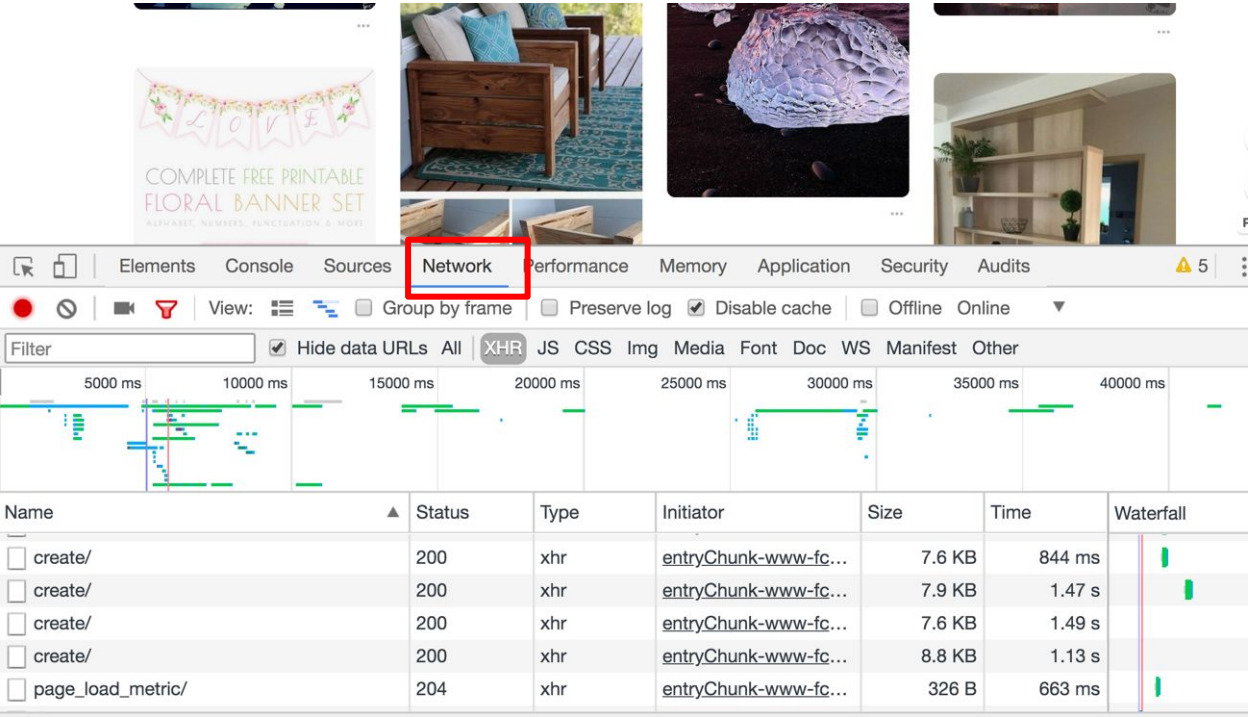
1. XMLHTTP Request

2. Fetch API

3. 3rd Party Libraries, jQuery

XHR Requests OK

Every request, makes an xhr request.



Official (Closed) Non-Sensitive

BASIC AJAX REQUEST

```
var XHR = new XMLHttpRequest();
XHR.onreadystatechange = function(){
  if(XHR.readyState == 4 && XHR.status == 200){
    console.log(XHR.responseText);
  }
};
XHR.open("GET", "https://api.github.com/zen");
XHR.send();
```

Value	State	Description
0	UNSENT	Client has been created. open() not called yet.
1	OPENED	open() has been called.
2	HEADERS_RECEIVED	send() has been called, and headers and status are available.
3	LOADING	Downloading; responseText holds partial data.
4	DONE	The operation is complete.

Reading Material: https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest/readyState

PROBLEMS WITH XHR

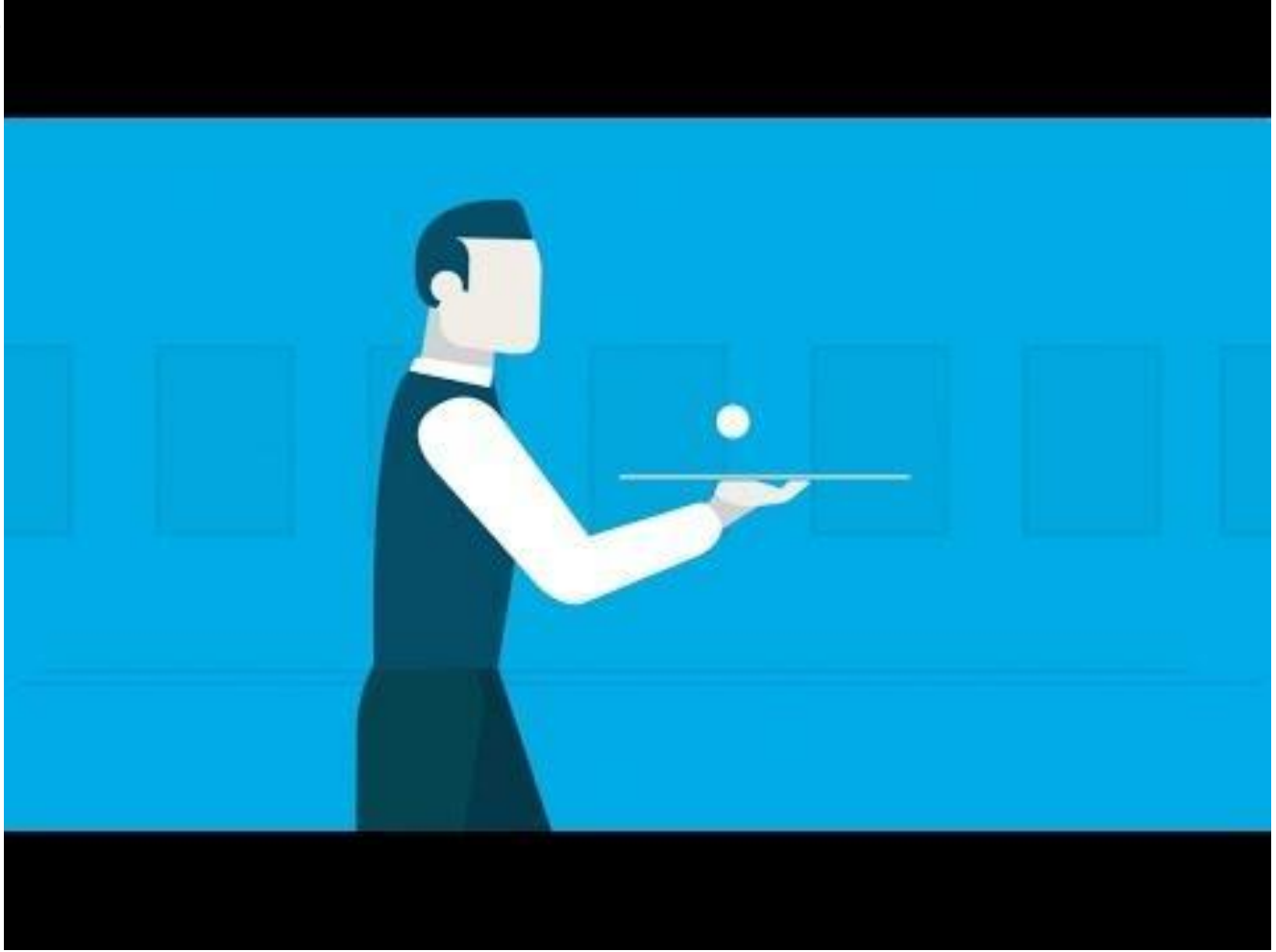
Ugly, bulky syntax
Very old method
No streaming allowed. Issue when large amounts of data comes in

API

Connecting with others
Application Interface Programming

#THEWAITER

What does he do?



Mulesoft: What is an API

https://www.youtube.com/watch?v=s7wmiS2mSXY

Take a request
Deliver it
Bring back a response

Source:
http://sproutsocial.com/insights/what-is-an-api/

Web Developers

Additional functionality and features to be integrated into the websites

Customisation, Flexibility, Mashups

Quicker, Faster (less complexity)

End Users

A good Web API can help increase the usefulness of a website and make the site interactive and more enjoyable..

Without API:

An app finds the current weather in London by opening http://www.weather.com/ and reading the webpage like a human does, interpreting the content.

With API:

An app finds the current weather in London by sending a message to the weather.com API (in a structured format like JSON). The weather.com API then replies with a structured response.

Reading Reference:
https://www.quora.com/What-is-an-API-4

Accessing API

Connecting with others
Application Interface Programming

Testing out API & JSON

We can try out the Github API by simply entering the github api url and our username

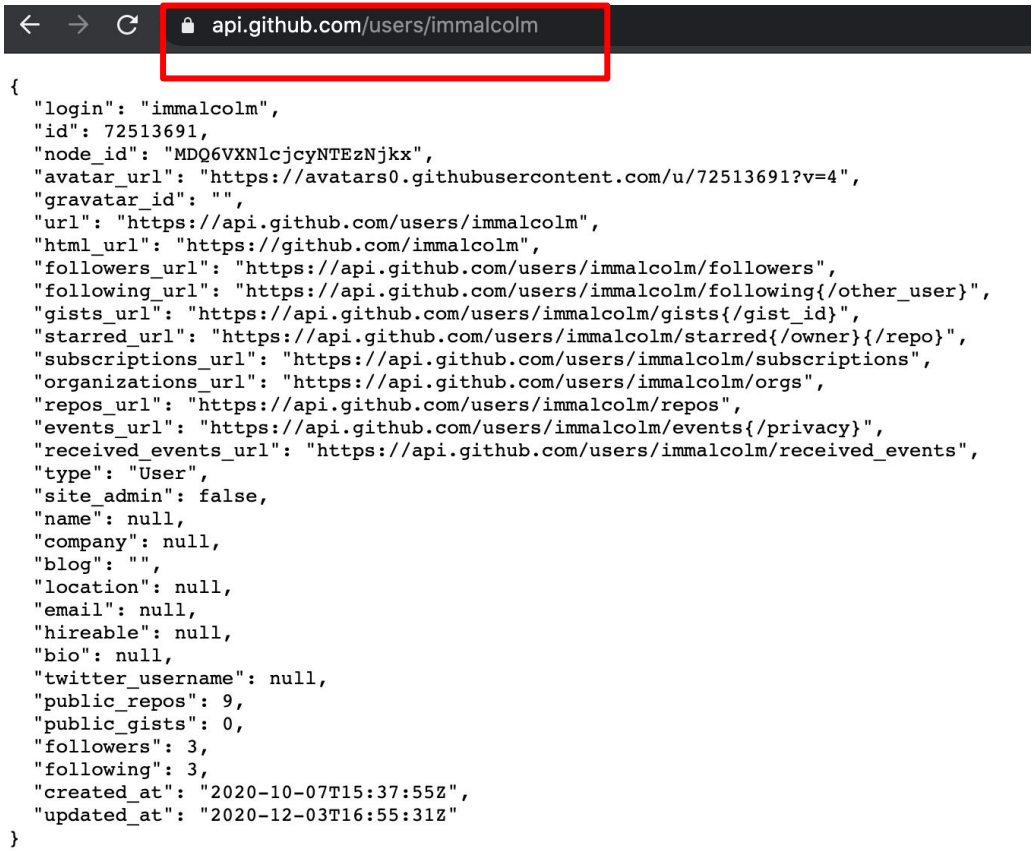
This API returns all the public information regarding the user.

Try it out on your own!

What gets returned is JSON string data. So we will need to call JSON.parse(data) to convert our JSON data into Javascript Objects.

Example of data stored into a data object using JSON.parse
https://replit.com/@immalcolm/simple-githubjson#script.js

https://api.github.com/users/immalcolm



POSTMAN

- ✔ Postman makes API development faster, easier, and better
- ✔ Highly used
- ✔ Intuitive user interface to send requests, save responses, add tests, and create workflows

https://www.postman.com/downloads/

Source: https://www.getpostman.com/

Create New | Templates | API Network

BUILDING BLOCKS

Request
Create a basic request

Collection
Save your requests in a collection for reuse and sharing

Environment
Save values you frequently use in an environment

ADVANCED

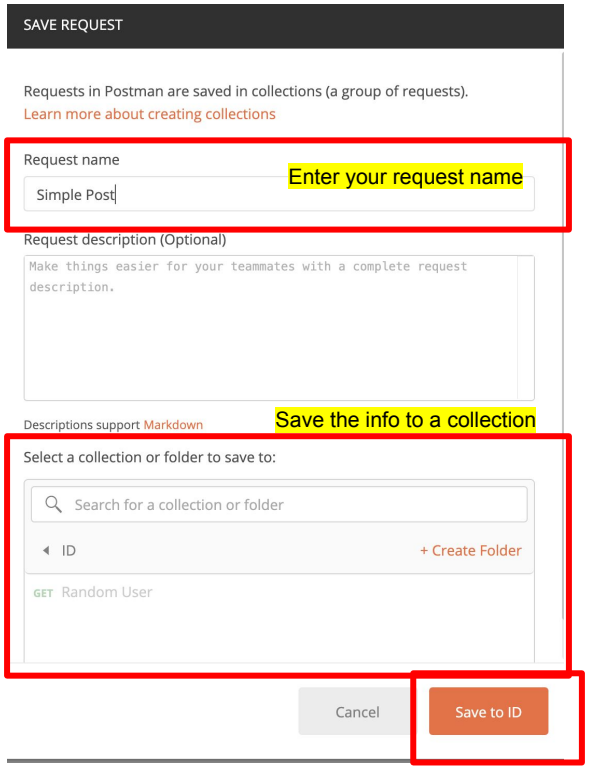
API Documentation
Create and publish beautiful documentation for your APIs

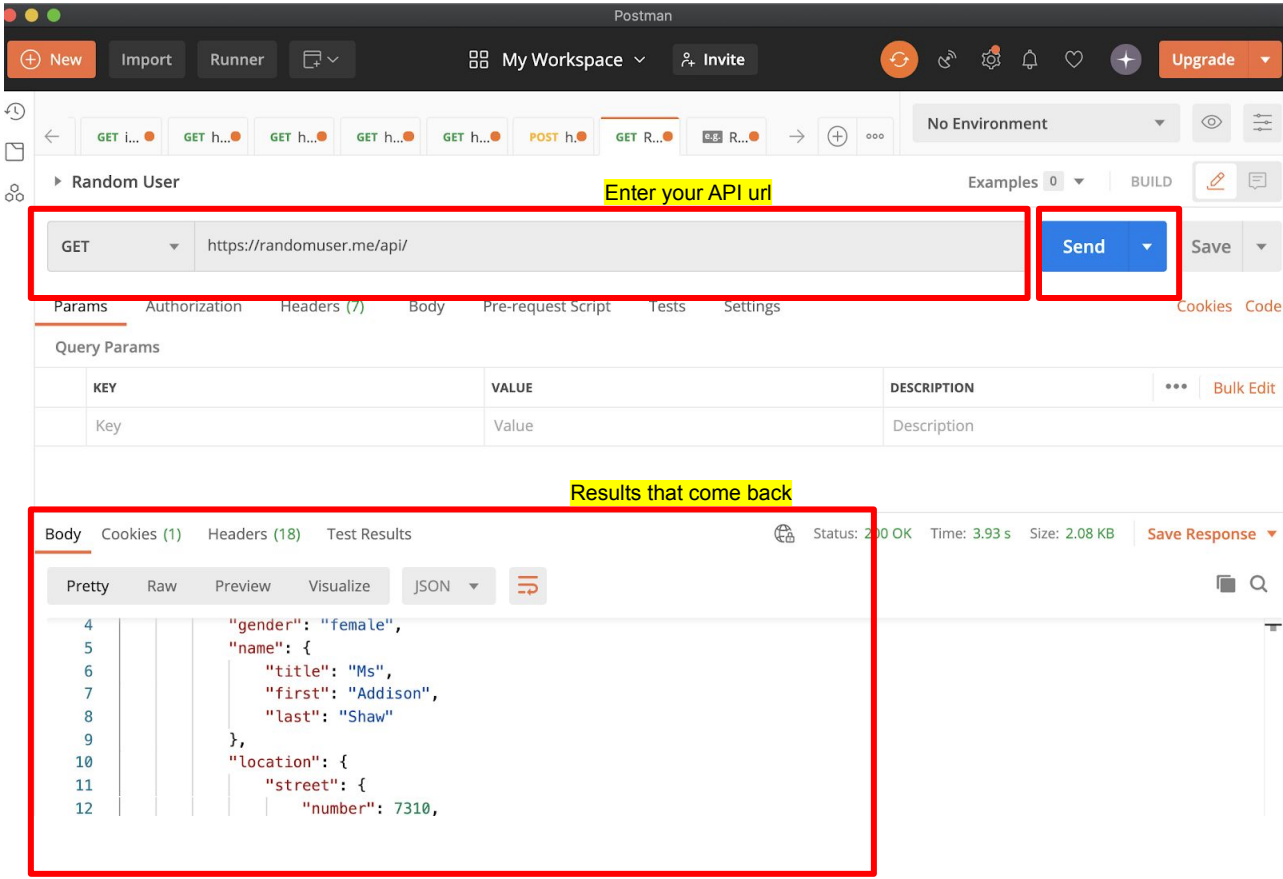
Mock Server
Create a mock server for your in-development APIs

Monitor
Schedule automated tests and check performance of your APIs

API
Manage all aspects of API design, development, and testing

Not sure where to start? Use a template to see how Postman can help you in your work.





ACTIVITY

“FIRST API REQUEST” (10mins)

#TODO
Download POSTMAN from http://getpostman.com
Follow instructions from your tutor
Resource: https://ipapi.co/

Create a GET request to https://ipapi.co/json to retrieve your IP address + location

Fetch

FETCH

```
fetch(url)
.then(function(res){
  console.log(res);
})
.catch(function(error
){
  console.log(error);
})
```

FETCH with Options

```
fetch(url, {
  method: 'POST',
  body: JSON.stringify({
  name: 'red',
  login: 'redmonster',
  })
})
.then(function (data) {
  //return data.json();
//do something
})
.then(function (data) {
  //process the return earlier
}).
.catch(function (error){
//handle error
})
```

https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch

FETCH

A basic fetch request is really simple to set up. Have a look at the following code:

```
fetch('http://example.com/movies.json')
  .then(response => response.json())
  .then(data => console.log(data));
```

Here we are fetching a JSON file across the network and printing it to the console. The simplest use of `fetch()` takes one argument — the path to the resource you want to fetch — and returns a promise containing the response (a `Response` object).

This is just an HTTP response, not the actual JSON. To extract the JSON body content from the response, we use the `json()` method (defined on the `Body` mixin, which is implemented by both the `Request` and `Response` objects.)

FETCH BROWSER SUPPORT

Browser compatibility

Update compatibility data on GitHub

	Chrome	Edge	Firefox	Internet Explorer	Opera	Safari	Android webview	Chrome for Android	Firefox for Android	Opera for Android	Safari on iOS	Samsung Internet
`fetch`	42	14	39	No	29	10.1	42	42	39	29	10.3	4.0
Support for blob: and data:	48	79	?	No	?	?	43	48	?	?	?	5.0
referrerPolicy	52	79	52	No	39	11.1	52	52	52	41	No	6.0
signal	66	16	57	No	53	11.1	66	66	57	47	11.3	9.0
Streaming response body	43	14	Yes	No	29	10.1	43	43	No	No	10.3	4.0

What are we missing?

- Full support
- No support
- Compatibility unknown
- Experimental. Expect behavior to change in the future.
- * See implementation notes.
- User must explicitly enable this feature.

Reading & Source: https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API

Let's Code #EXERCISE

Object Exercise

See repl.it Project :

AJAX Random user


```
{
  "results": [
    {
      "gender": "male",
      "name": {
        "title": "mr",
        "first": "brad",
        "last": "gibson"
      },
      "location": {
        "street": "9278 new road",
        "city": "kilcoole",
        "state": "waterford",
        "postcode": "93027",
        "coordinates": {
          "latitude": "20.9267",
          "longitude": "-7.9310"
        },
        "timezone": {
          "offset": "-3:30",
          "description": "Newfoundland"
        }
      },
      "email": "brad.gibson@example.com",
      "login": {
        "uuid": "155e77ee-ba6d-486f-95ce-0e0c0fb4b919",
        "username": "silverswan131",
        "password": "firewall",
        "salt": "TQA1Gz7x",
        "md5": "dc523cb313b63dfe5be2140b0c05b3bc",
        "sha1": "7a4aa07d1bedcc6bcf4b7f8856643492c191540d",
        "sha256": "74364e96174afa7d17ee52dd2c9c7a4651fe1254f471a78bda0190135dcd34
      },
      "dob": {
        "date": "1993-07-20T09:44:18.674Z",
        "age": 26
      },
      "registered": {
        "date": "2002-05-21T10:59:49.966Z",
        "age": 17
      },
      "phone": "011-962-7516",
      "cell": "081-454-0666",
      "id": {
        "name": "PPS",
        "value": "0390511T"
      },
      "picture": {
        "large": "https://randomuser.me/api/portraits/men/75.jpg",
        "medium": "https://randomuser.me/api/portraits/med/men/75.jpg",
        "thumbnail": "https://randomuser.me/api/portraits/thumb/men/75.jpg"
      },
      "nat": "IE"
    }
  ],
  "info": {
    "seed": "fea8be3e64777240",
    "results": 1,
    "page": 1,
    "version": "1.3"
  }
}
```


JSON set information
Results set

JSON set information
Info set

https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch

PSI

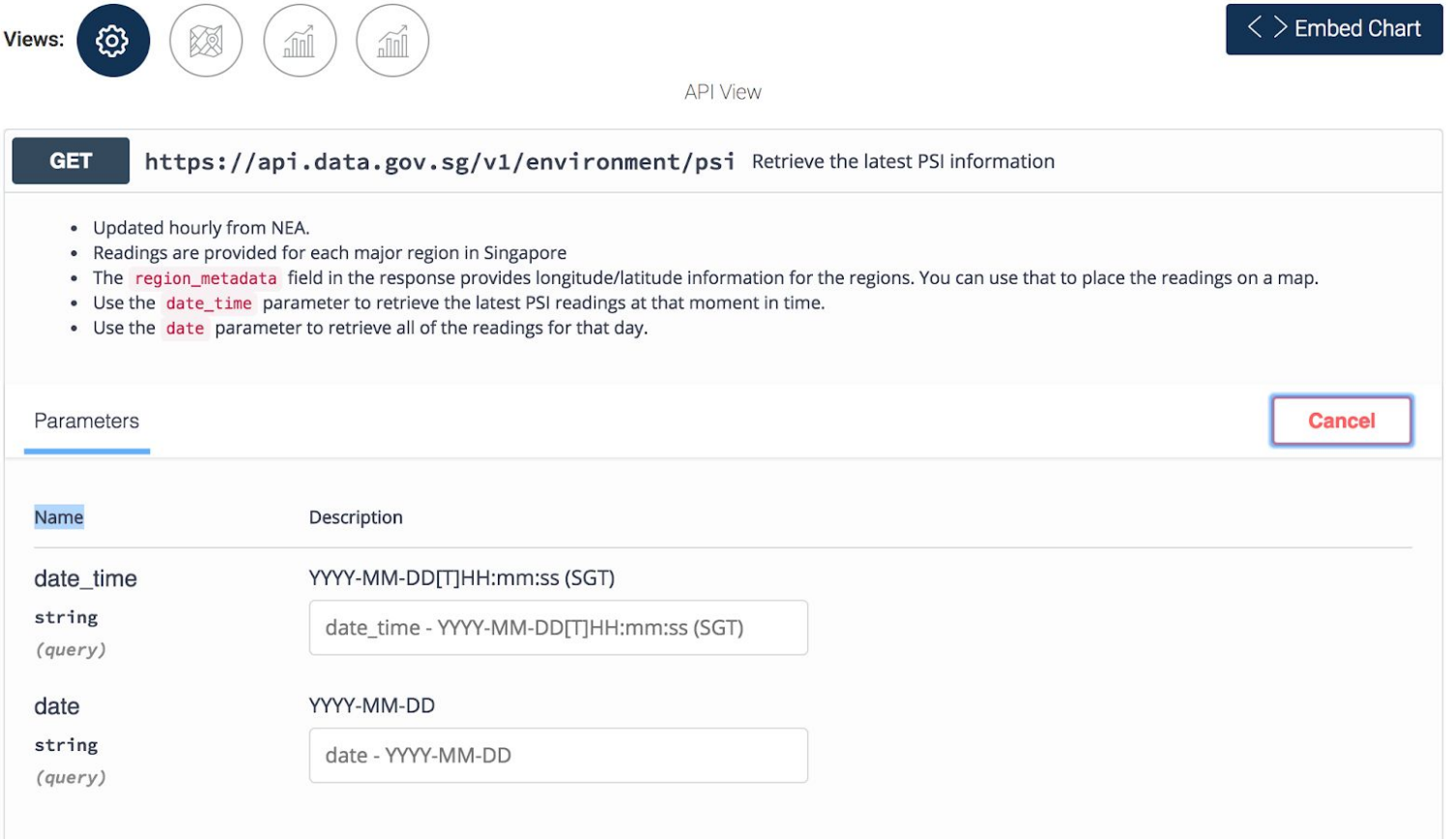
#pollutionstandardindex
#hazeliterally

Singapore at a glance

Singapore Residents By Ethnic Group, End June, Annual (2018)

Domestic Exports By Area (2014)

Source: https://data.gov.sg/



Source: https://data.gov.sg/dataset/psi

Let's Code #EXERCISE

10mins

Object Exercise

See repl.it Project :

AJAX PSI

PSI 24h	PSI 3h
national: 57 south: 55 north: 57 east: 51 central: 53 west: 47	national: 53 south: 48 north: 59 east: 52 central: 43 west: 53

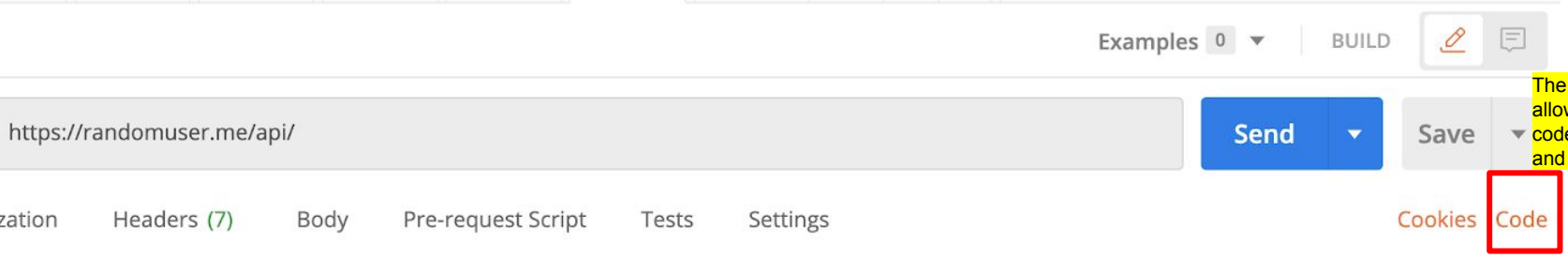
https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch

API

Key Takeaway?

Practise & Practise

It takes quite a bit of trial and error with APIs. Always test using POSTMAN first. This is to allow code to be tested first before you actually start coding it.



The POSTMAN code feature allows you to show the AJAX code for JS Fetch, jQuery and many more. Try it out :)

Important Readings

AJAX
https://applyhead.com/xmlhttprequest-vs-fetch-api/
https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest
https://www.w3schools.com/jquery/jquery_ajax_get_post.asp

API Reference
https://www.programmableweb.com/
WATCH THIS: https://www.youtube.com/watch?v=7YcW25PHnAA

JSON Refresher
https://www.youtube.com/watch?v=wI1CWzNtE-M
https://www.youtube.com/watch?v=rJesac0_Ftw

Tools
http://crossorigin.me/
JSONView Chrome Extension
https://jsonformatter.org/
Animista.net (CSS Animation Tool)

READING

“Asynchronous” means that when a client requests data from a web server, it doesn’t freeze until the server replies. On the contrary, the user can still navigate the pages. As soon as the server returns a response, a relevant function manipulates the returned data behind the scenes.

“JavaScript” is the language which instantiates an AJAX request, parses the corresponding AJAX response, and finally updates the DOM.

A client uses the XMLHttpRequest or XHR API to make a request to a server. Think of the API (Application Programming Interface) as a set of methods which specify the rules of communication between the two interested parties. However, note that the incoming data from an AJAX request can be in any format and not only in XML format.

READING

Learning Above Event Callbacks

https://www.youtube.com/watch?v=8aGhZQkoFbQ&vl=en