Advanced AJAX

Web Development Boot Camp

Lesson 6.2

API Recap

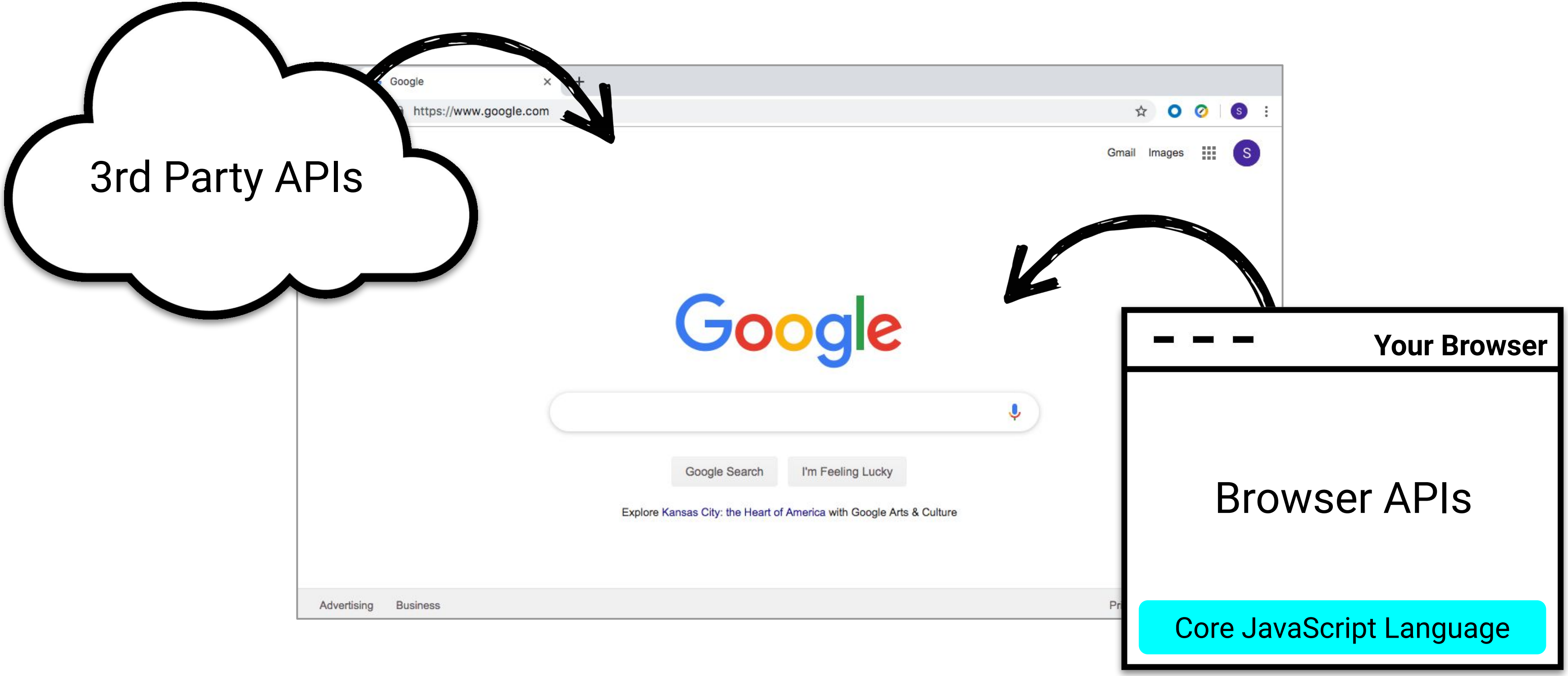
What Is an API?

> An **application programming interface (API)** is a set of features and rules that exist inside a software program (the application) enabling interaction with it through software—as opposed to a human user interface.

— MDN Web Docs, developer.mozilla.org

API Recap

In software development, APIs are often the bridge between different components.



3rd Party APIs

Your Browser

Browser APIs

Core JavaScript Language

API Use Cases

Three Common Use Cases for APIs

01 Provide pre-built code for accessing and sending data to a centralized database (e.g., weather data, movie data from IMDB).

02 Provide pre-built code for creating or utilizing other software components (e.g., Google Maps, Spotify tools).

03 Interface with physical sensors or hardware devices (e.g., Nest thermostats, Philips Hue).

API

Use Case #1:

Accessing and Sending Data

Example:

Airbnb utilizes the Google Maps API to power its entire mapping service.

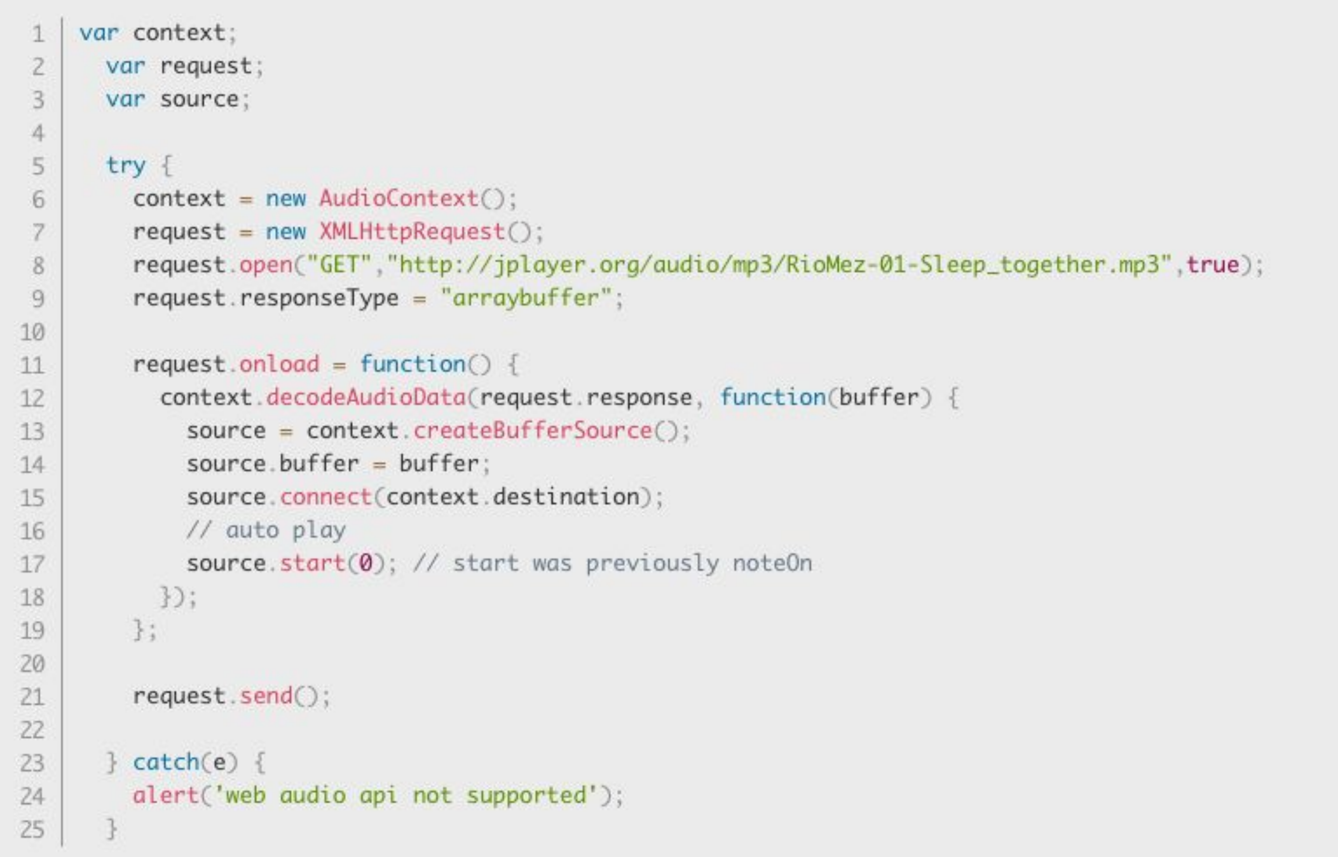
API

API

Use Case #2:

Utilizing Pre-Built Code

Use Case #2: Utilizing Pre-Built Code



```
var context;
  var request;
  var source;

  try {
    context = new AudioContext();
    request = new XMLHttpRequest();
    request.open("GET","http://jplayer.org/audio/mp3/RioMez-01-Sleep_together.mp3",true);
    request.responseType = "arraybuffer";

    request.onload = function() {
      context.decodeAudioData(request.response, function(buffer) {
        source = context.createBufferSource();
        source.buffer = buffer;
        source.connect(context.destination);
        // auto play
        source.start(0); // start was previously noteOn
      });
    };

    request.send();

  } catch(e) {
    alert('web audio api not supported');
  }
```

developer.mozilla.org

Use Case #3:

Controlling Physical Hardware

Nest Smart Thermostat API: developer.nest.com

Philips Hue API: developers.meethue.com

NodeBots (Cylon.js) API: cylonjs.com

Getting Data

Steps to use server-side API

01 Determine the URL Endpoint

02 Obtain permission to use the service (API or APP key)

03 Choose a way to make an AJAX call

04 Identify any needed query parameters

05 Make the asynchronous request

06 Decode the response

Determine the URL Endpoint

1. Find an API that you want to use
2. Read their documentation

   Need:
   *** URL**
   *** HTTP method**

Access Permission: API (or App) Keys

These are like passwords

Protect them!

- They identify you, so they are your responsibility
- Sometimes they are connected to your credit card
- Do not check them into your repo
- Client side protection:
  - Adding it to your app is publishing it to the world
  - Some free services don’t care
  - Some pay services require you to “white-list” deployment domains
    - What is a domain? E.g., **mywebsite.com**
- Server side protection:
  - Environment variables
    - `.env` files
    - Server dashboard

Choose a way to make an AJAX call

There are multiple ways: some suiggestions

1. Browser object `XMLHttpRequest`
   https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest

   Original method. Difficult to use, more code to setup and handle responses. Some browser differences early on.

2. Browser `fetch` API (has cross browser
   https://medium.com/beginners-guide-to-mobile-web-development/the-fetch-api-2c962591f5c

   Newer native method. Some browser incompatibilities

3. `axios` js: 3rd party library that wraps `XMLHttpRequest`

   npm package and browser cdn

4. `$.ajax`: jQuery’s ajax method (wraps `XMLHttpRequest`)

jquery

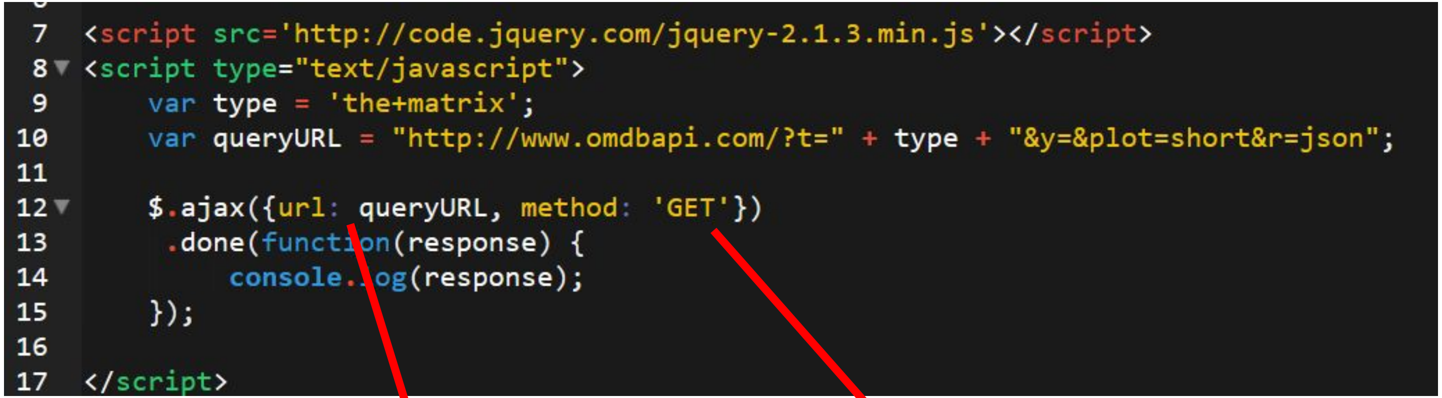
What **jQuery method** do we use to retrieve data from a server-side API?

AJAX

What **two parameters** do we pass into AJAX to retrieve data from online?

AJAX

What two parameters do we pass into AJAX to retrieve data from online?



```
<script src='http://code.jquery.com/jquery-2.1.3.min.js'></script>
<script type="text/javascript">
    var type = 'the+matrix';
    var queryURL = "http://www.omdbapi.com/?t=" + type + "&y=&plot=short&r=json";

    $.ajax({url: queryURL, method: 'GET'})
     .done(function(response) {
         console.log(response);
    });

</script>
```

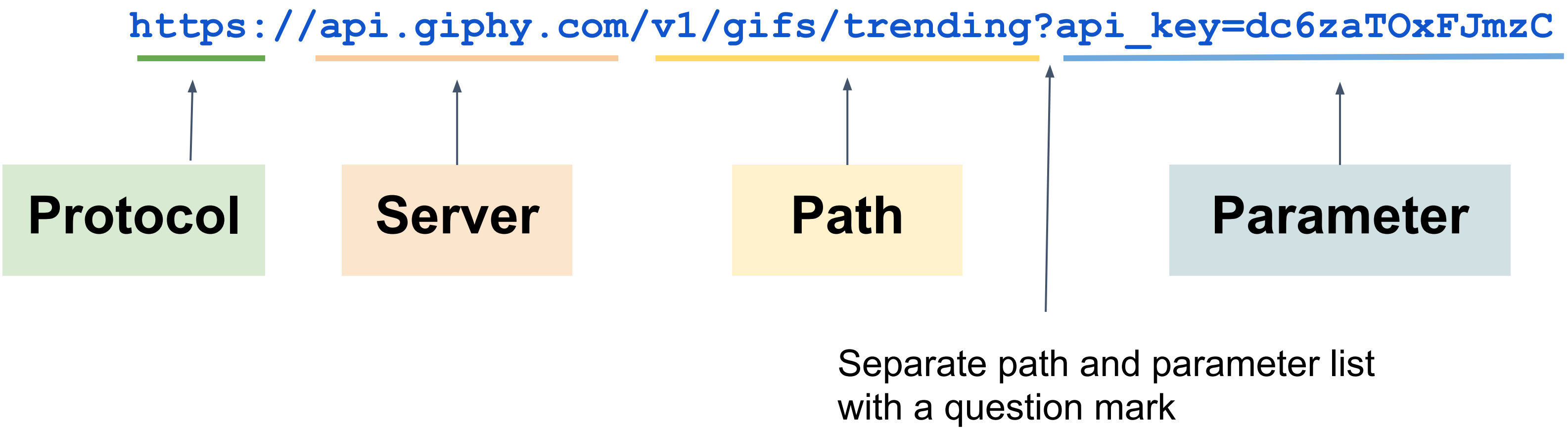
`url`

`‘GET’` method

Identify query parameters

What are they? Read the API's documentation for that endpoint

GET parameters: go on the URL



`https://api.giphy.com/v1/gifs/trending?api_key=dc6zaTOxFJmzC`

Protocol

Server

Path

Parameter

Separate path and parameter list
with a question mark

A parameter is a name=value pair: `api_key=dc6zaTOxFJmzC`

Multiple parameters

Separated by an `&`

Example:

2 parameters

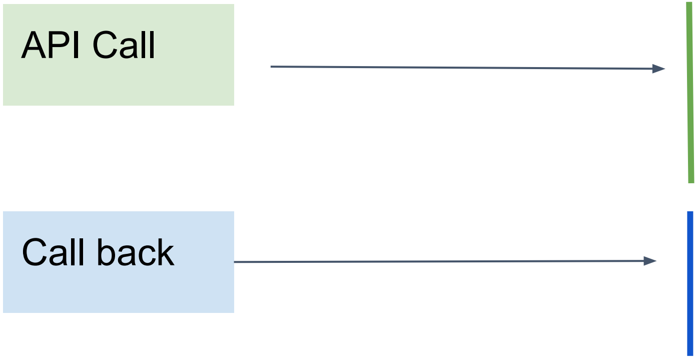
My `api_key` parameter is `dc6zaTOxFJmzC`

My query parameter, `q` is `cheeseburgers`

`https://api.giphy.com/v1/gifs/search?api_key=dc6zaTOxFJmzC&q=cheeseburgers`

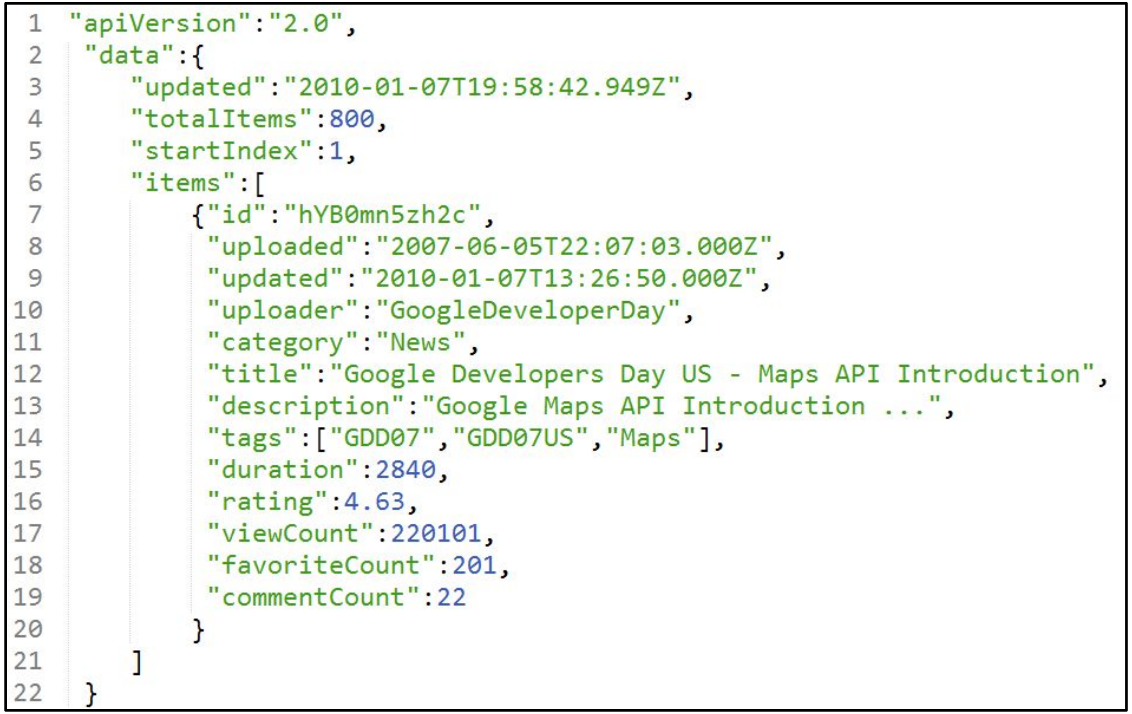
Make the API call

It is like a phone call that requests a call back



```
$.ajax({
  url: queryURL,
  method: "GET"
}).then(function(response) {
  console.log(response);
});
```

Reponse: a JSON Object



```
"apiVersion":"2.0",
 "data":{
    "updated":"2010-01-07T19:58:42.949Z",
    "totalItems":800,
    "startIndex":1,
    "items":[
        {"id":"hYB0mn5zh2c",
         "uploaded":"2007-06-05T22:07:03.000Z",
         "updated":"2010-01-07T13:26:50.000Z",
         "uploader":"GoogleDeveloperDay",
         "category":"News",
         "title":"Google Developers Day US - Maps API Introduction",
         "description":"Google Maps API Introduction ...",
         "tags":["GDD07","GDD07US","Maps"],
         "duration":2840,
         "rating":4.63,
         "viewCount":220101,
         "favoriteCount":201,
         "commentCount":22
        }
    ]
 }
```

What’s in the Response?

Two techniques

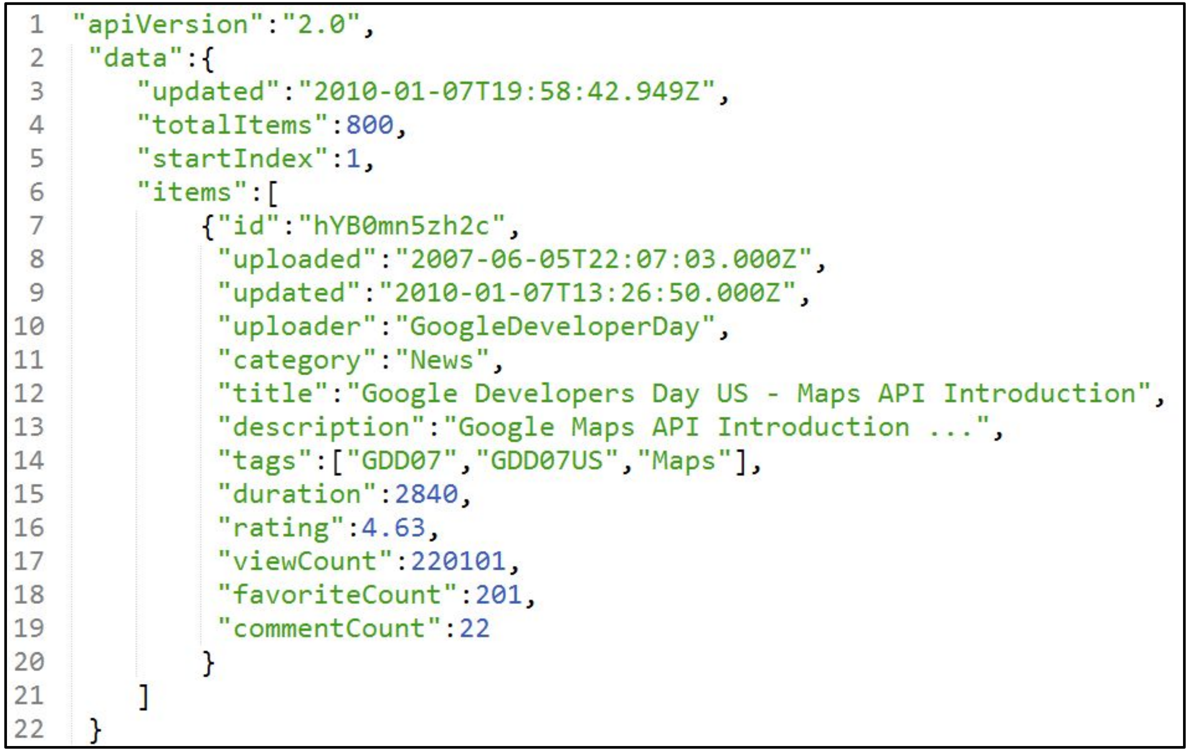
1. **Read the documentation (can be spotty or missing)**
2. **Console log**

What Is JSON?

JSON stands for **JavaScript Object Notation** and is nothing more than simple JavaScript objects used as a **data interchange format.**

JSON Recap

JSON is a lightweight data interchange format used to correlate keys with values.

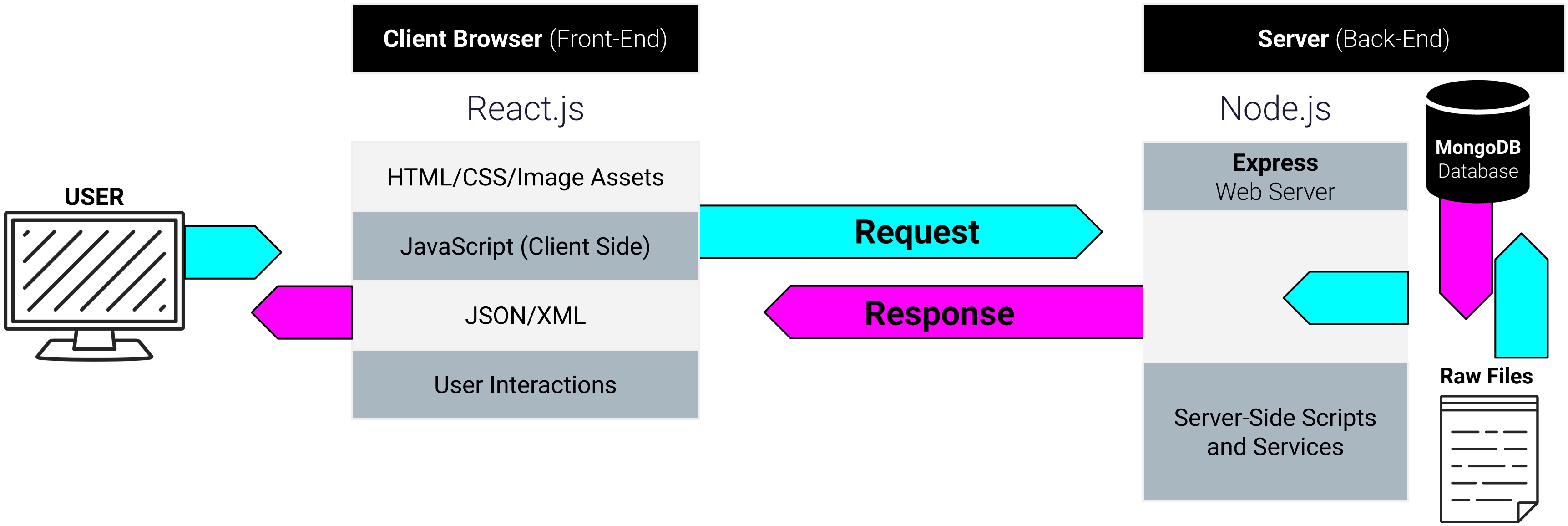


```json
"apiVersion":"2.0",
 "data":{
    "updated":"2010-01-07T19:58:42.949Z",
    "totalItems":800,
    "startIndex":1,
    "items":[
        {"id":"hYB0mn5zh2c",
         "uploaded":"2007-06-05T22:07:03.000Z",
         "updated":"2010-01-07T13:26:50.000Z",
         "uploader":"GoogleDeveloperDay",
         "category":"News",
         "title":"Google Developers Day US - Maps API Introduction",
         "description":"Google Maps API Introduction ...",
         "tags":["GDD07","GDD07US","Maps"],
         "duration":2840,
         "rating":4.63,
         "viewCount":220101,
         "favoriteCount":201,
         "commentCount":22
        }
    ]
 }
```

Knowledge Is Power

Welcome to Full-Stack Development

Full-stack development refers to the development of *every* aspect of the web application, from visuals and interactions to data transfer and processing.



AJAX and API Overview

At this point, you should know:

01 The definition of API

02 The definition of JSON

03 Why AJAX methods are used (i.e., retrieving data in databases).

04 How to create a basic AJAX get request using jQuery (i.e., include URL and ‘`GET`’).

Questions?