Lecture 10: JSON, AJAX, Third-Party Web APIs

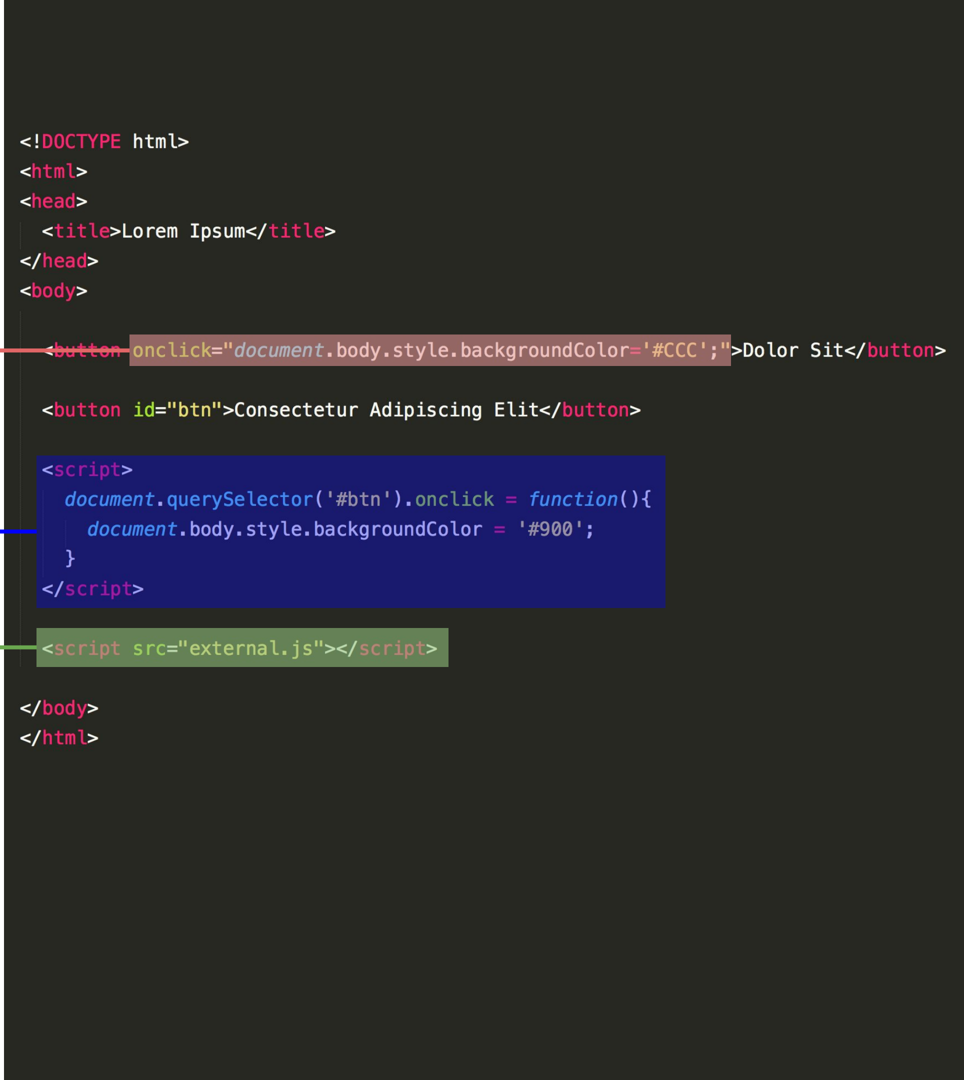
ITP 303 Full-Stack Web Development

External JS

JavaScript Types

Just like CSS, there are 3 types of JavaScript:

1. Inline
2. Internal
3. External



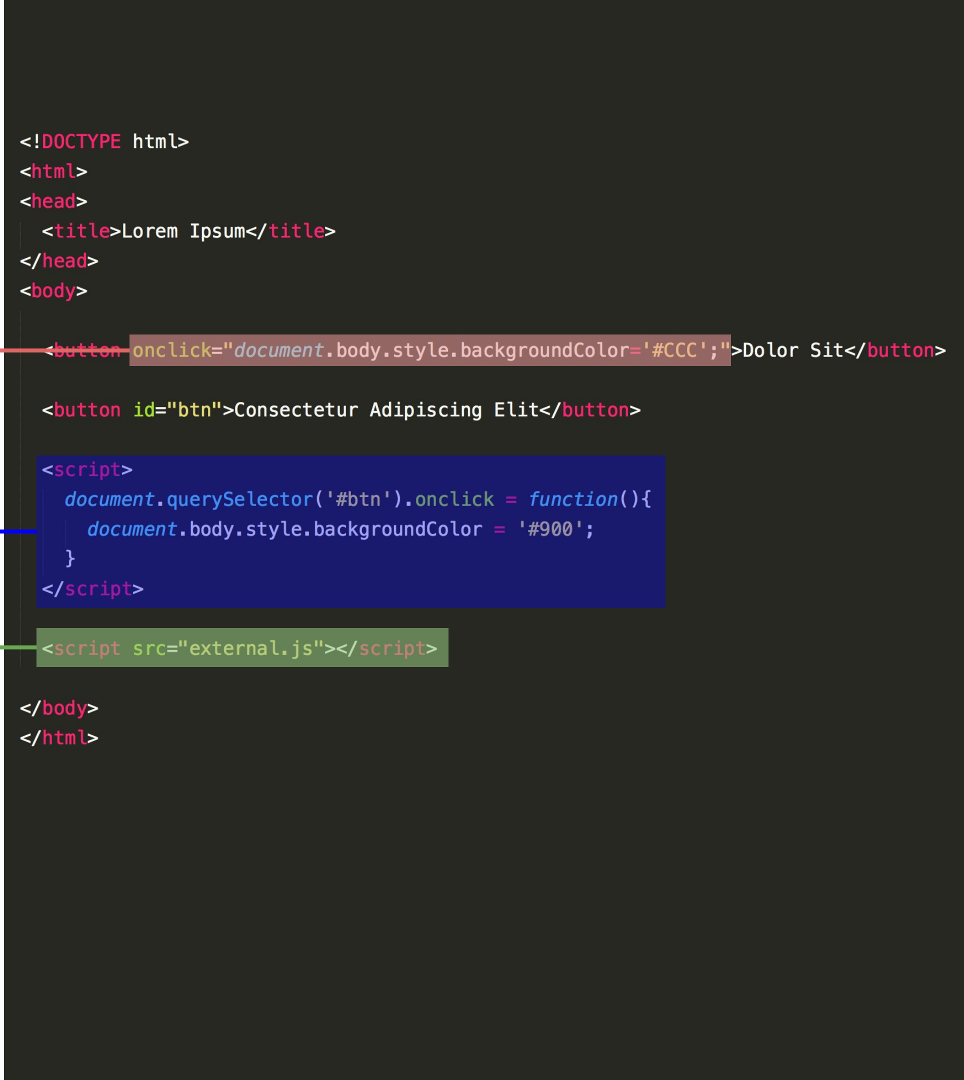
```html
<!DOCTYPE html>
<html>
<head>
  <title>Lorem Ipsum</title>
</head>
<body>

  <button onclick="document.body.style.backgroundColor='#CCC';">Dolor Sit</button>

  <button id="btn">Consectetur Adipiscing Elit</button>

  <script>
    document.querySelector('#btn').onclick = function(){
      document.body.style.backgroundColor = '#900';
    }
  </script>

  <script src="external.js"></script>

</body>
</html>
```

JavaScript Types

Just like CSS, there are 3 types of JavaScript:

1. ~~Inline~~
2. Internal
3. External

Just like inline CSS, avoid using inline JS.

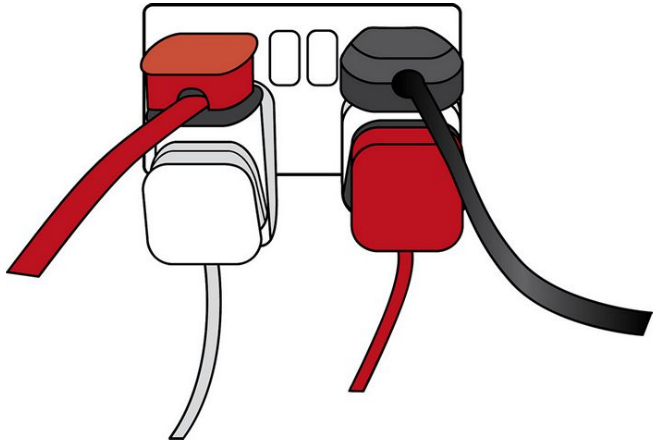


```html
<!DOCTYPE html>
<html>
<head>
  <title>Lorem Ipsum</title>
</head>
<body>

  <button onclick="document.body.style.backgroundColor='#CCC';">Dolor Sit</button>

  <button id="btn">Consectetur Adipiscing Elit</button>

  <script>
    document.querySelector('#btn').onclick = function(){
      document.body.style.backgroundColor = '#900';
    }
  </script>

  <script src="external.js"></script>

</body>
</html>
```

APIs

Application Programming Interface (API)

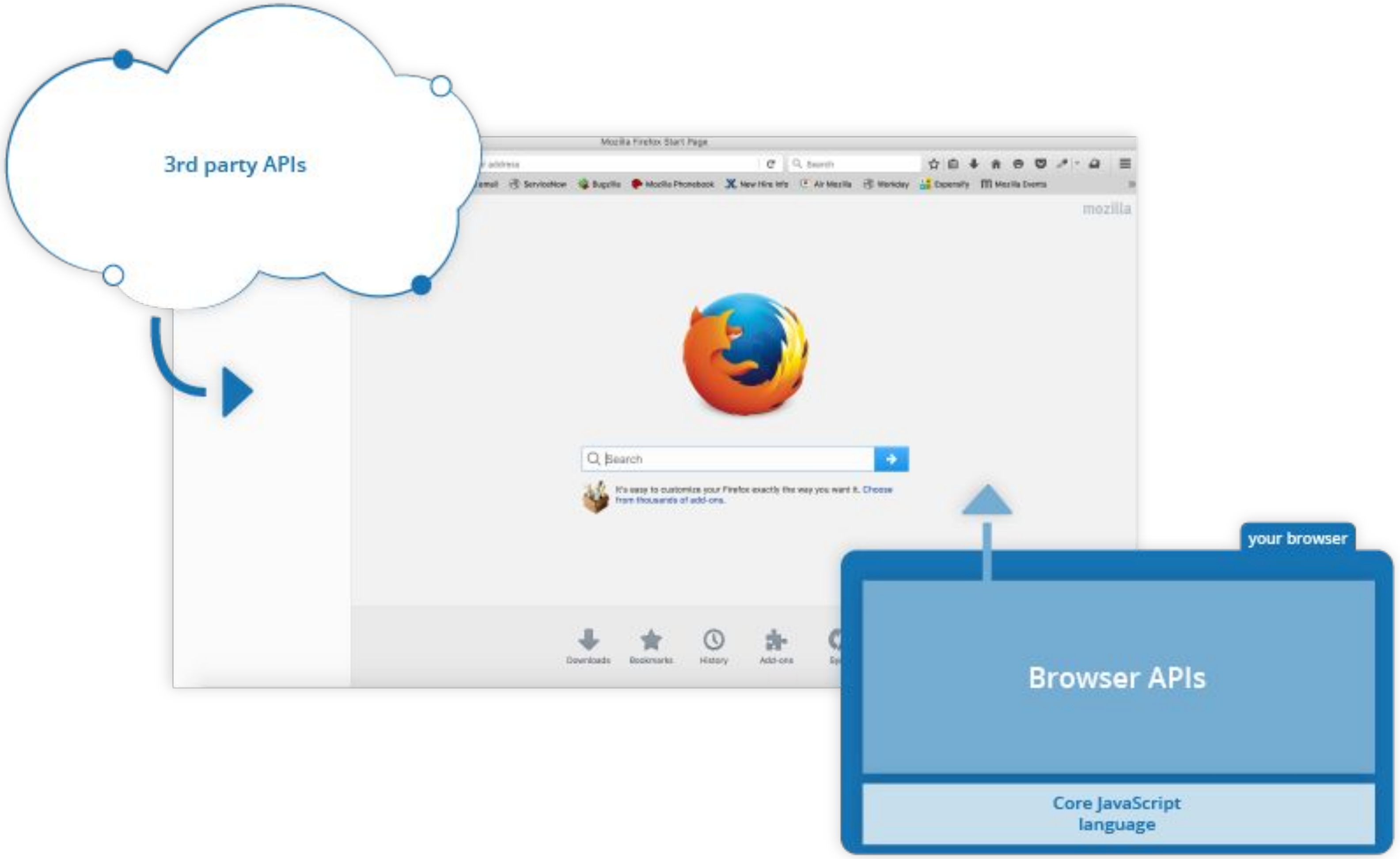
- Constructs made available in programming languages to allow developers to create complex functionality more easily
- They abstract more complex code away from you, providing some easier syntax to use in its place
- Real world example: plug socket



You’ve been using APIs actually

- console.log() - https://developer.mozilla.org/en-US/docs/Web/API/Console
- document.querySelector() - https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model
- Others include: Video, Audio, and Canvas (for drawing) APIs

Browsers have their own API that allows us to implement things. It sits on top of JavaScript.


3rd party APIs
Mozilla Firefox Start Page
mozilla
Search
It's easy to customize your Firefox exactly the way you want it. Choose from thousands of add-ons.
Downloads
Bookmarks
History
Add-ons
your browser
Browser APIs
Core JavaScript language

Third-Party APIs

- Other platforms (e.g. Twitter, Facebook) create APIs for developers to access their functionality in their web pages.
- Examples:
    - Add latest tweets from Taylor Swift on my website
    - Allow users to login to my website with their facebook account
    - Display today’s headlines from New York Times on my website
    - Send text messages from my website using the Twilio API

How do we access Third Party APIs?

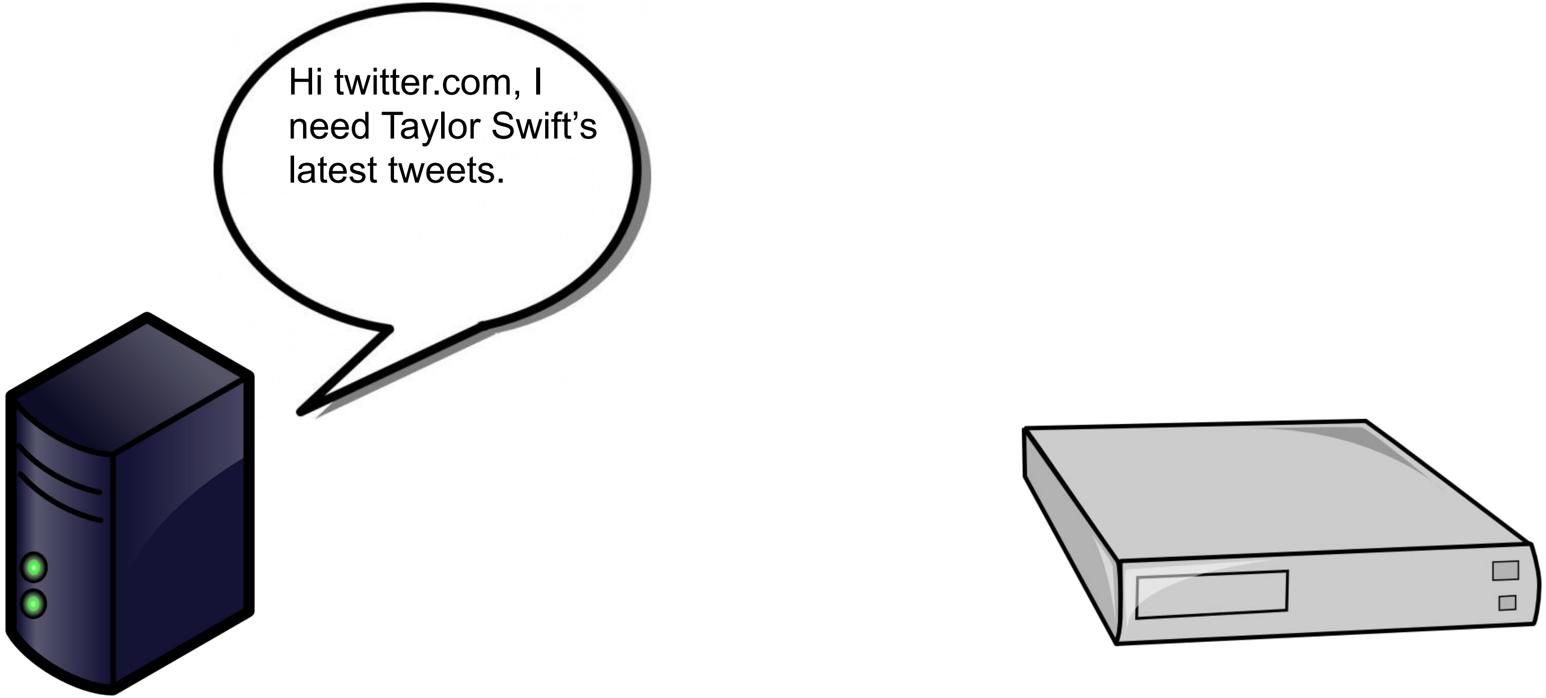
1. Obtain an API key.

2. Link to their JavaScript library using the <script> tag
   OR
   make HTTP requests to specific URLs

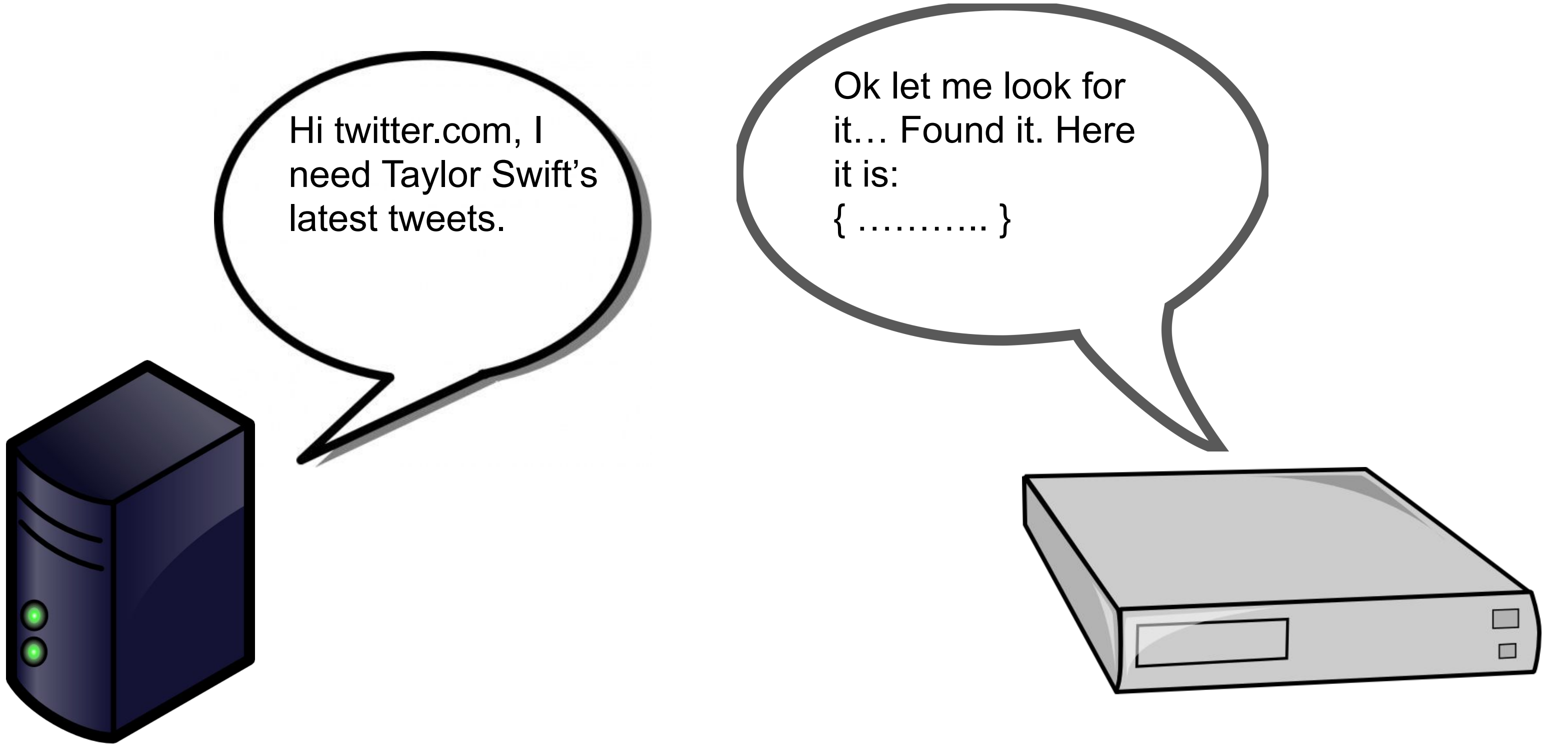
RESTful API

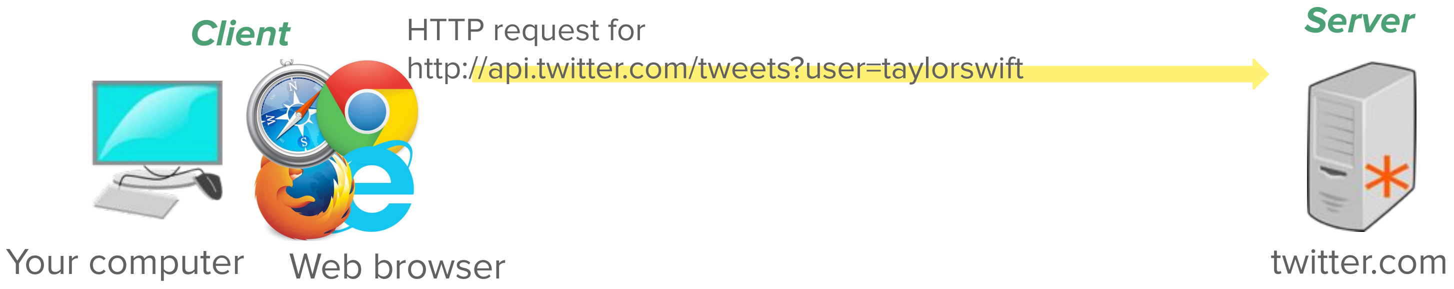
- REST - Representational State Transfer
- A set of rules that developers follow when they create their API.
  - Grammar of APIs
- Most commonly used APIs are REST APIs these days

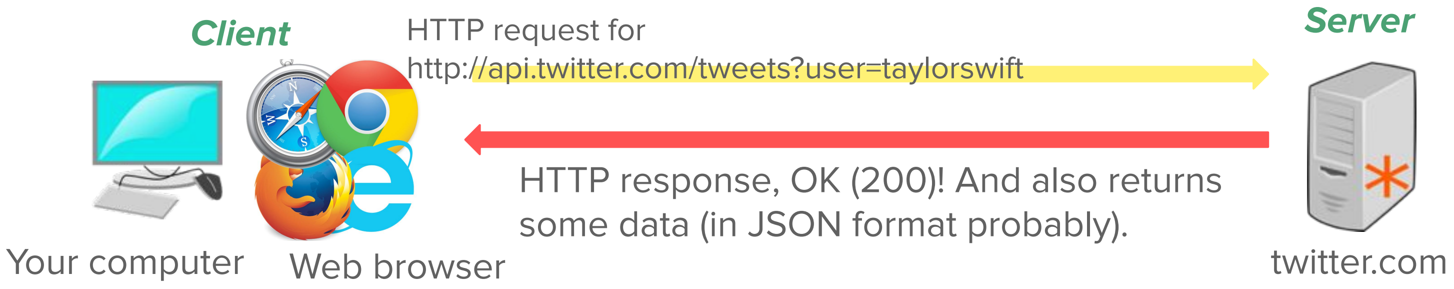
Making a HTTP request



Receiving a response


Hi twitter.com, I need Taylor Swift's latest tweets.
Ok let me look for it… Found it. Here it is: { ……….. }


Client
HTTP request for
http://api.twitter.com/tweets?user=taylorswift
Server
Your computer
Web browser
twitter.com



HTTP request for
http://api.twitter.com/tweets?user=taylorswift

HTTP response, OK (200)! And also returns some data (in JSON format probably).

The Anatomy of a Request

Endpoint

```
http://api.twitter.com/tweets
```

The Anatomy of a Request

Endpoint

```
http://api.twitter.com/tweets
```

Parameters

```
http://api.twitter.com/tweets?user=taylorswift&limit=20&lang=en
```

The Anatomy of a Request

Endpoint

```
http://api.twitter.com/tweets
```

Parameters

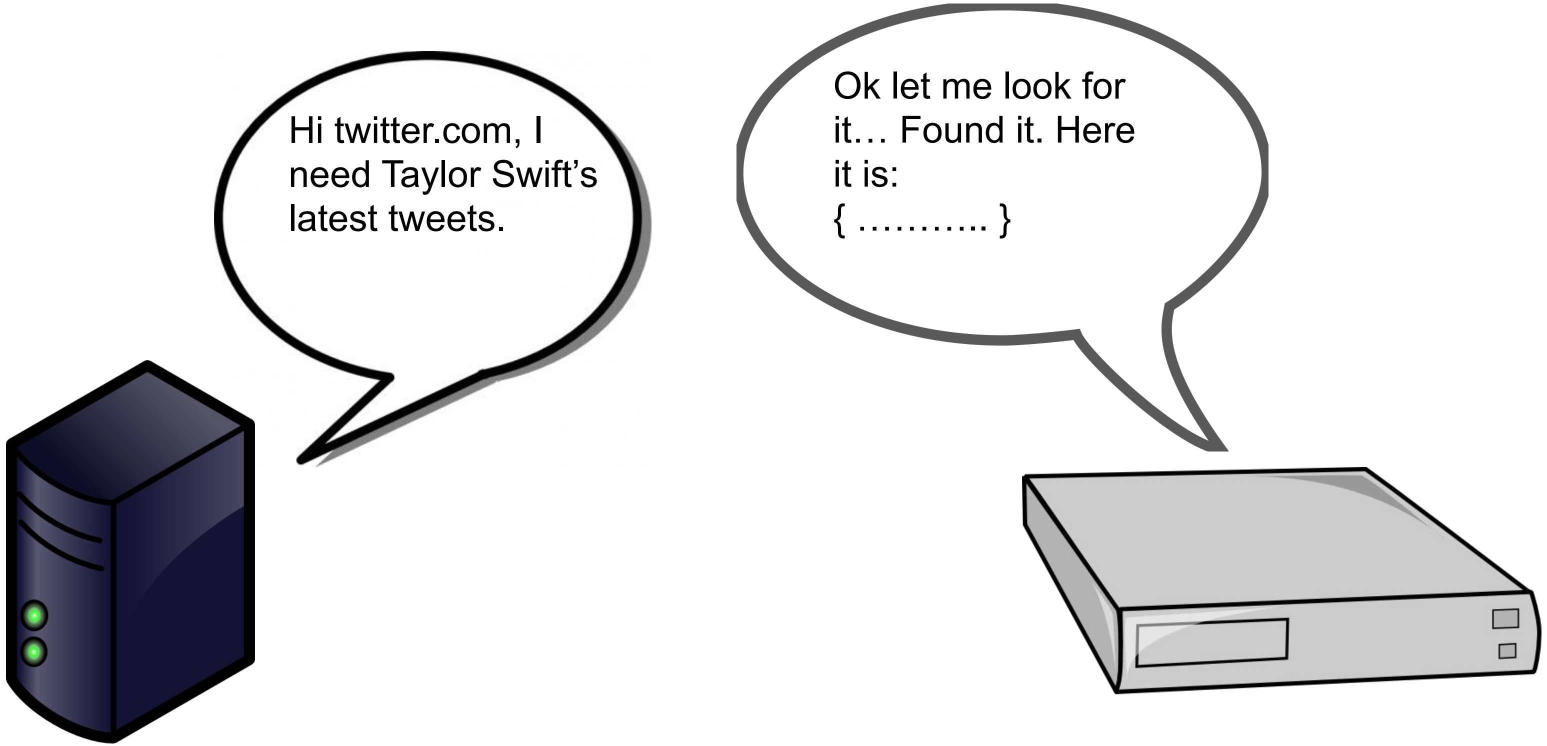
```
http://api.twitter.com/tweets?user=taylorswift&limit=20&lang=en
```

Method

```
GET http://api.twitter.com/tweets?user=taylorswift&limit=20&lang=en
```

```
POST http://api.twitter.com/tweets?user=taylorswift&limit=20&lang=en
```

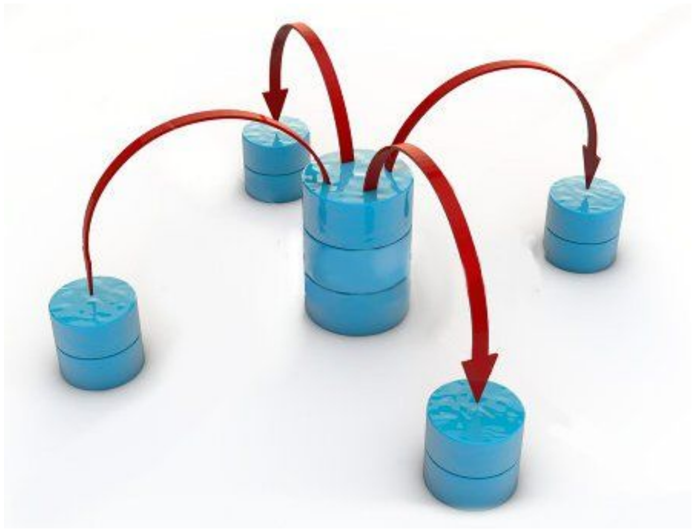
Receiving a response


Hi twitter.com, I need Taylor Swift's latest tweets.
Ok let me look for it… Found it. Here it is: { ……….. }

JSON

Sharing is caring

- We live in a world where tons and tons of data is collected and used.
- There was a need to somehow standardize and share this data across different devices



JavaScript Object Notation (JSON)

- A lightweight data format
- Easy for humans to read and write
- Easy for machines to parse and generate
- Derived from JavaScript
- Language independent
    - Not ONLY for JavaScript. Can be used in other applications (Java, Objective C, Swift, etc)

JavaScript Objects vs JSON

```javascript
var person = {
    firstName: 'Tommy',
    lastName: 'Trojan',
    email: 'ttrojan@usc.edu',
    phone: {
        home: '123-123-1234',
        cell: '456-456-4567'
    },
    pets: [
        'Spot',
        'Bolt'
    ]
};
```

```javascript
var person = {
    "firstName": "Tommy",
    "lastName": "Trojan",
    "email": "ttrojan@usc.edu",
    "phone": {
        "home": "123-123-1234",
        "cell": "456-456-4567"
    },
    "pets": [
        "Spot",
        "Bolt"
    ]
};
```

JavaScript Objects vs JSON

- JSON must use **double quotes** for keys and values
- JSON cannot contain methods - strictly data only
- Otherwise looks very similar to JavaScript objects

JSON string

```
"{
    "firstName": "Tommy",
    "lastName": "Trojan",
    "email": "ttrojan@usc.edu",
    "phone": {
        "home": "123-123-1234",
        "cell": "456-456-4567"
    },
    "pets": [
        "Spot",
        "Bolt"
    ]
}"
```

Sometimes JSON is given to you as a **string**, so have to parse the string into JS objects by using:

`JSON.parse()`

JSON methods

`JSON.parse()`	Parses a JSON string, constructing the JavaScript value or object described by the string.
`JSON.stringify()`	Converts a JavaScript value to a JSON string.

To recap...

- Many platforms provide some kind of **API** that allows our web application to interact with that platform
  - E.g. Twitter's API allows us to get Taylor Swift's latest tweets.
- To interact with an API, we can (usually) make a HTTP **request** to an **endpoint** and if successful, we will get some kind of **response** back.
- One of the common formats the response we receive back is **JSON**.
  - JSON can be easily converted into JS objects so that's great!

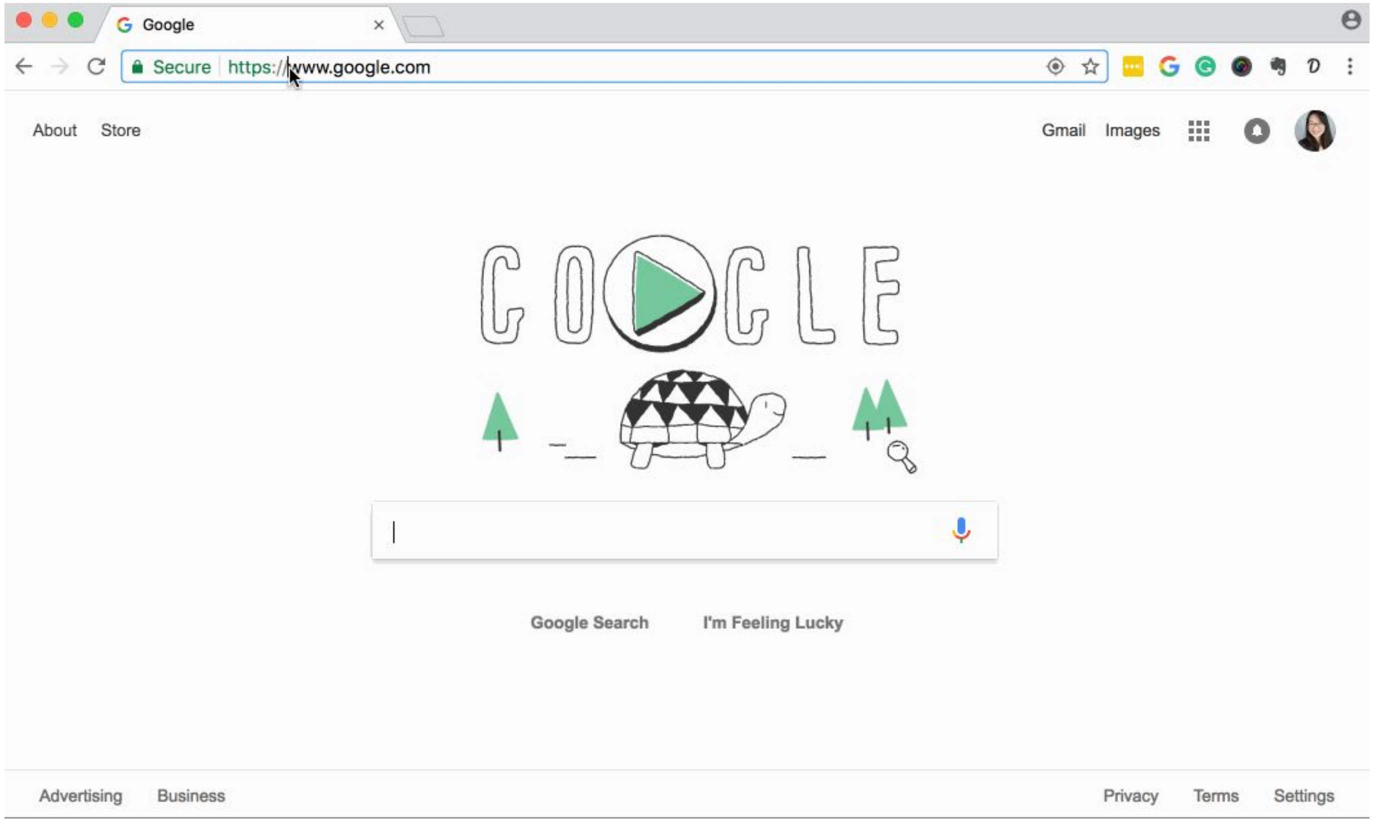
To recap...

- Many platforms provide some kind of **API** that allows our web application to interact with that platform.
  - E.g. Twitter's API allows us to get Taylor Swift's latest tweets.
- To interact with an API, we can (usually) make a HTTP **request** to an **endpoint** and if successful, we will get some kind of **response** back.
- One of the common formats the response we receive back is **JSON**.
  - JSON can be easily converted into JS objects so that's great!

That's great! But now how can I make HTTP requests?

AJAX

HTTP requests typically require a browser refresh



AJAX - **A**synchronous **J**avaScript **A**nd **X**ML

- Allows browsers to make requests to the server **without reloading the page**
- Uses the **XMLHttpRequest** object to communicate to servers
- Can send and receive information in various formats such as JSON, XML, HTML, and text files (not limited to XML like its name)

Same-origin policy

- Security measure
- Web browsers limit script access/interaction between different domains

Exceptions to the same-origin policy

- <img> src attribute
  - `<img src=”http://itpwebdev.com/images/a1.jpg”>`
- <link rel=”stylesheet”> href attribute
  - `<link rel=”stylesheet” href=”http://itpwebdev.com/css/style.css”>`
- <script> src attribute
  - `<script src=”http://itpwebdev.com/js/myscript.js”>`

Exceptions to the same-origin policy

- <img> src attribute
  - `<img src=”http://itpwebdev.com/images/a1.jpg”>`
- <link rel=”stylesheet”> href attribute
  - `<link rel=”stylesheet” href=”http://itpwebdev.com/css/style.css”>`
- <script> src attribute
  - `<script src=”http://itpwebdev.com/js/myscript.js”>`
- **CORS** (Cross-origin resource sharing) enabled resources