Introduction to AJAX

What is an API?

What is an API?

- **A**pplication **P**rogramming **I**nterface (**API**)
- Software that allows two programs to communicate with each other
    - It all starts with shared data
- The principle of API abstraction allows for decoupling applications
- Can be private or public

What is an API?

- **Application**
    - Any application
- **Programming**
    - The engineering part that translates given inputs into outputs
- **Interface**
    - The interface, the way we interact

What is an API?

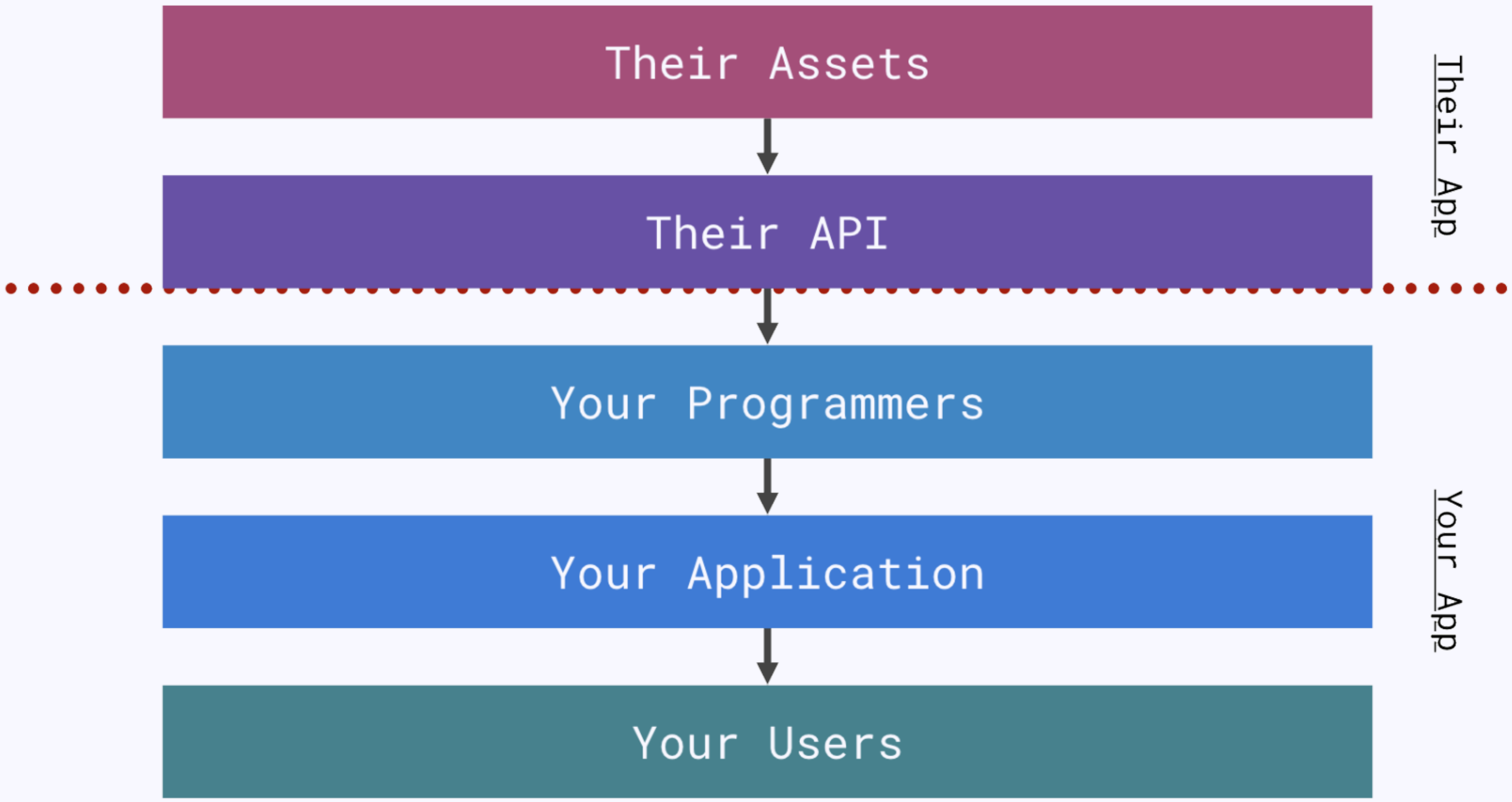
Any interface that software can use to access some piece of currency (e.g. data, processes etc.)

They are a standard way of communicating between systems, that hides the complexities

Types of APIs

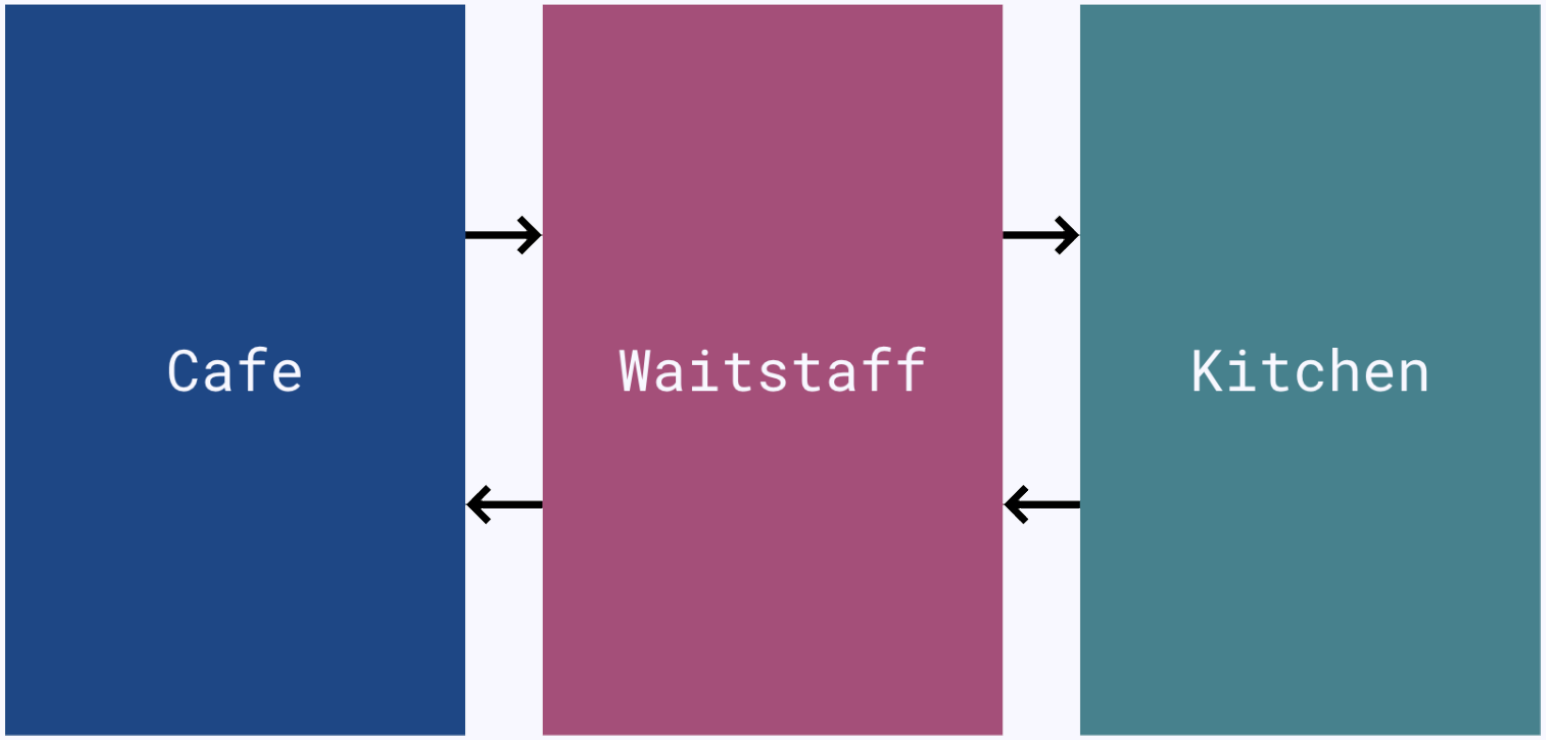
- Operating Systems
- Remote APIs
- Libraries, Frameworks and Software Development Kits (we will focus on this)
- Web APIs (we will focus on this)

How do APIs work?


Their Assets
Their API
Your Programmers
Your Application
Your Users
Their App
Your App

The Five Elements

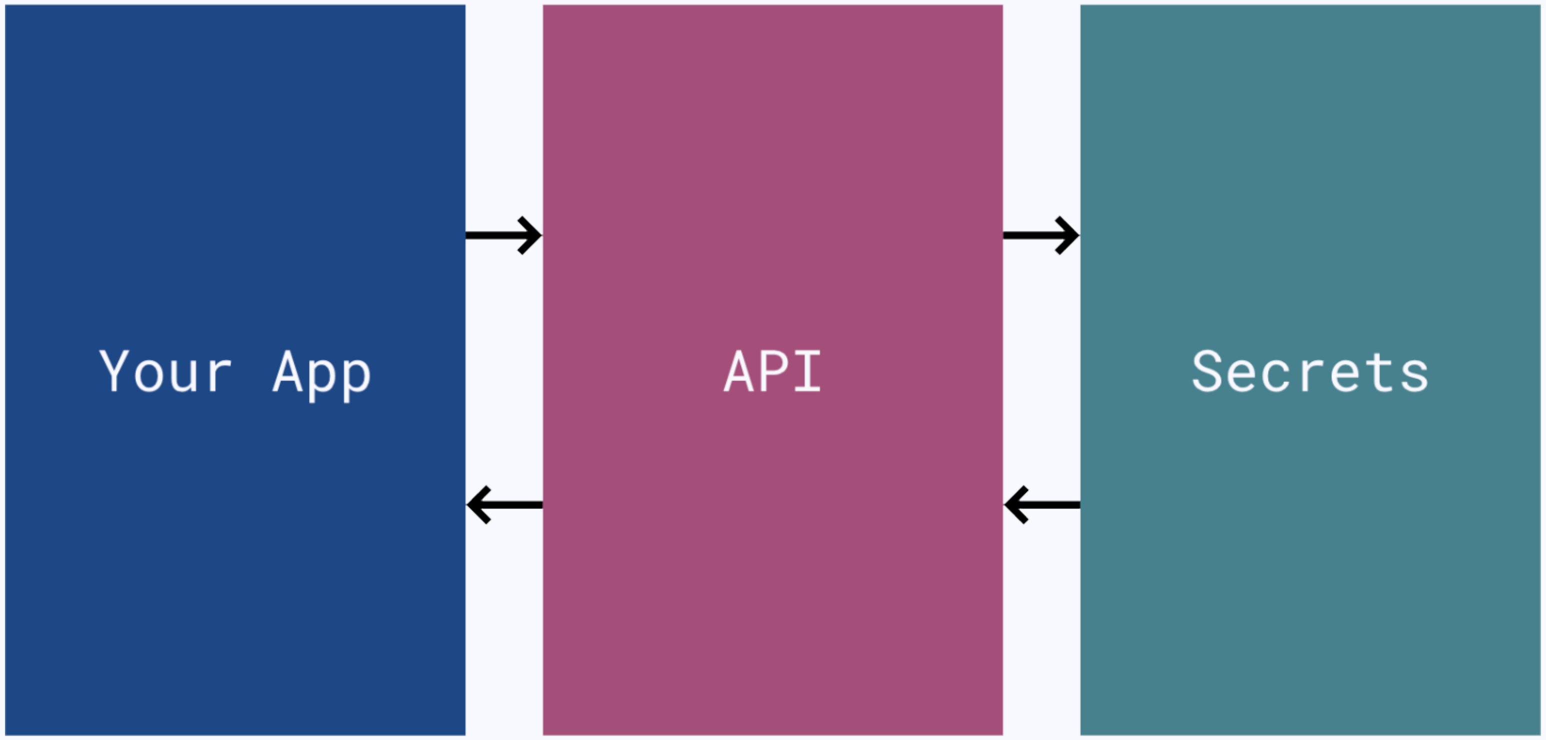
- **Assets** - Anything that is chosen to be shared (data, processes etc.)
- **API** - The gateway to those assets
- **Developers** - The API is exposed to your developers
- **Your Application** - Your developers code your application, and it is powered by the API
- **Your Users** - Your users use the app that is created



Cafe

Waitstaff

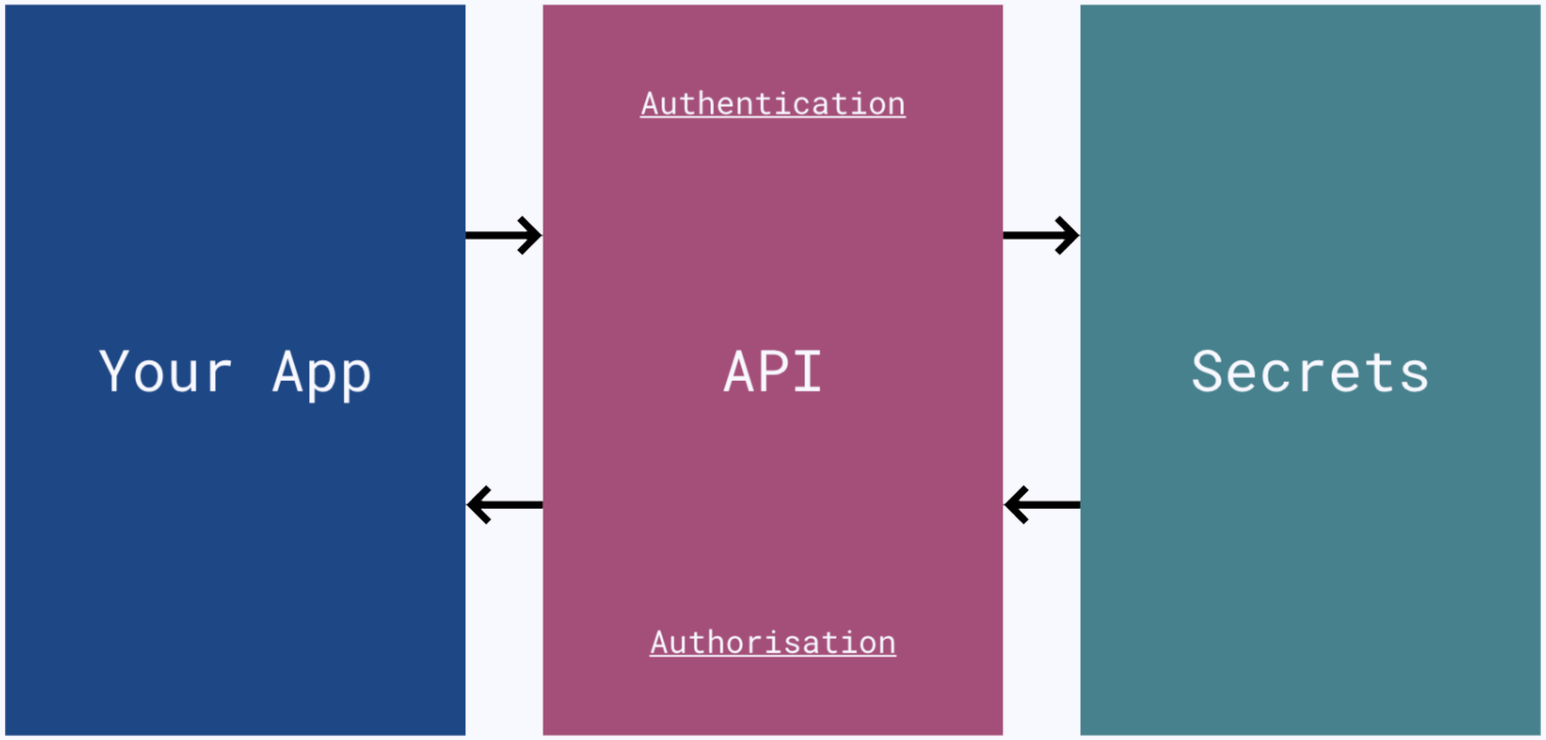
Kitchen



Your App

API

Secrets


Your App
Authentication
API
Authorisation
Secrets

APIs

Authentication

Are you allowed to access the system?

Authorisation

What can you access in the system?

Often we need to sign up to APIs

Web APIs

To use these:

- We use the web!
- We make an HTTP request
    - Providing all necessary information - often in JSON format
- The API performs all of their fancy stuff
- The API returns the data as an HTTP response
- We use the data

But how does the web work

HTTP Methods

GET - Asking for data

PUT - Updating data

POST - Creating data

DELETE - Deleting data

Plus more!

Libraries, Frameworks & SDKs

To use these:

- You either download & include the code
- Or it is provided by the platform
    - e.g. browsers provide functionality for all sorts of things (video, audio, 3D stuff etc.)

A Web Platform Sojourn

What are some of the latest APIs that the web platform is providing?

- Payment Request API
- Generic Sensor API
- WebGL
- WebVR
- Speech Recognition and Speech Synthesis
- Progressive Web Apps
- Notifications API
- Plus, many more...

Benefits of APIs

Benefits for providers

- APIs let you build one app off another
- APIs are like a universal plug
- APIs can be profitable
- APIs can allow you to decouple code - can make your applications more performant and finding developers easier
- APIs can essentially outsource complexity
- API providers can promote your application

Benefits for consumers

- APIs can allow you to get data, and add features, that would otherwise be difficult and time-consuming
- APIs can make your app realtime
- APIs can introduce similar flows to an application (e.g. logins etc.)
- Apps using APIs can provide useful starting data for users

Downsides

- Building them
- Maintenance
- Hosting
- Documentation
- For developers - future support
- Reliance on other services

Applications of APIs

Some Applications

- Uber
- Google Assistant / Siri / Alexa
- Xero
- Meetup
- General Assembly
- Panda
- Plus your app!

API Resources

API Resources

- Todd Motto: Public APIs
- APIList.fun
- Any API
- APIGee
- Postman
- Introduction to OAuth

AJAX

What is AJAX?

- Asynchronous JavaScript & eXtensible Markup Language
- It is a way to make your pages live
- You can talk to other servers while you are still on the page
- It is a technique to send and retrieve information behind the scenes without needing to refresh the page

Where is AJAX used?

- In feeds (such as twitter)
- Chat rooms and messaging apps
- For voting and rating
- Autocompletion
- Form submission and validation
- To access data that you don't have
- To show extra information
- In games (e.g. to save scores)

Why is AJAX so good?

- It makes your pages live
- It is much faster
- It tends to give a greater user experience
- It is fancy
- It is popular in the workplace
- It is the foundation of things such as APIs

How to work with it?

- XMLHttpRequest
- Fetch

One Thing!

- To make API requests, we need a server (AJAX doesn't work on a file URL)
- Install HTTP-Server
- Restart the terminal
- To start the server, run the following command:

  -
    ```
    http-server .
    ```

Fetch

What is Fetch?

- It is a way to make AJAX Requests
- It is a function that is defined automatically for us
- It is supported by all major browsers now
    - For those that don't support it, there is a polyfill

What is Fetch?

- You make an *HTTP Request* with it
    - Specifying the URL, the method etc.
- It comes back with an *HTTP Response*
    - Most of the time, the data is returned as **JSON**
- Then returns a *Promise*
    - We can work with the returned data in a `.then`

Using Fetch

```
fetch( URL, HTTP_OPTIONS? )
  .then( SUCCESS_HANDLER )
  .catch( ERROR_HANDLER );
```

Using Fetch

```javascript
const baseURL = "http://api.open-notify.org/astros";

fetch(baseURL, { method: "GET", dataType: "JSON" })
  .then(function(response) {
    return response.json();
  })
  .then(function(data) {
    console.log(data);
  });
```

Using Fetch

- We have make a Request to the API
- We parse the Response and turn it into a JS Object with `.json()`
- We can then work with the data!

Using Fetch

```
const url = "https://randomuser.me/api/?results=10";

fetch(url, { method: "GET", dataType: "JSON" })
  .then(function(res) {
    return res.json();
  })
  .then(function(data) {
    console.log(data);
  });
```

Using Fetch

```javascript
const url = "https://randomuser.me/api/?results=10&gender=male";

fetch(url, { method: "GET", dataType: "JSON" })
  .then(function(res) {
    return res.json();
  })
  .then(function(data) {
    console.log(data);
  });
```

OMDB API

- Go to OMDB APi's Website
- Get an API Key from here and select Free
- Input your details
- Check your email
- Click the verify API key link
- Copy your API key from that email

OMDB API - Authentication

```
const baseURL = "http://api.openweathermap.org/data/2.5/weather";
const parameters = "?q=Sydney&units=metric&appid=API_KEY";

fetch(baseURL + parameters)
  .then(function(response) {
    return response.json();
  })
  .then(function(data) {
    console.log(data);
  });
```

OpenWeatherMap API

- Go to the OpenWeatherMap API website
- Sign up for an API key here
- Fill in your details
- Log in
- Go to the API Key Tab on your settings page
- Copy the API Key
- It'll take ten minutes for the API Key to work

OpenWeatherMap API

```
const baseURL = "http://api.openweathermap.org/data/2.5/weather";
const parameters = "?q=Sydney&units=metric&appid=API_KEY";

fetch(baseURL + parameters)
  .then(function(response) {
    return response.json();
  })
  .then(function(data) {
    console.log(data);
  });
```

Resources

- MDN: Using Fetch
- CSS Tricks: Using Fetch
- Scotch.io: Fetch
- David Walsh: Fetch
- Google Developers: Fetch
- Google Developers: Working with the Fetch API
- MDN: Fetch API

Homework

- Get comfortable with ES2015
    - ES6.io
    - Babel.JS
- Find another API and do something cool with it!
    - e.g. ISS Location and Google Maps API
    - e.g. Google Books
    - Or one from here (if it says OAuth, don't pick it for the moment)