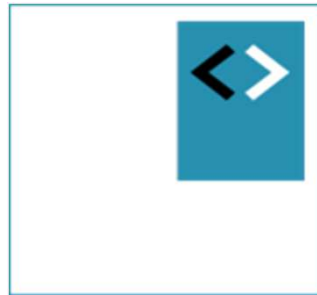


React Fundamentals

Module – Http communication



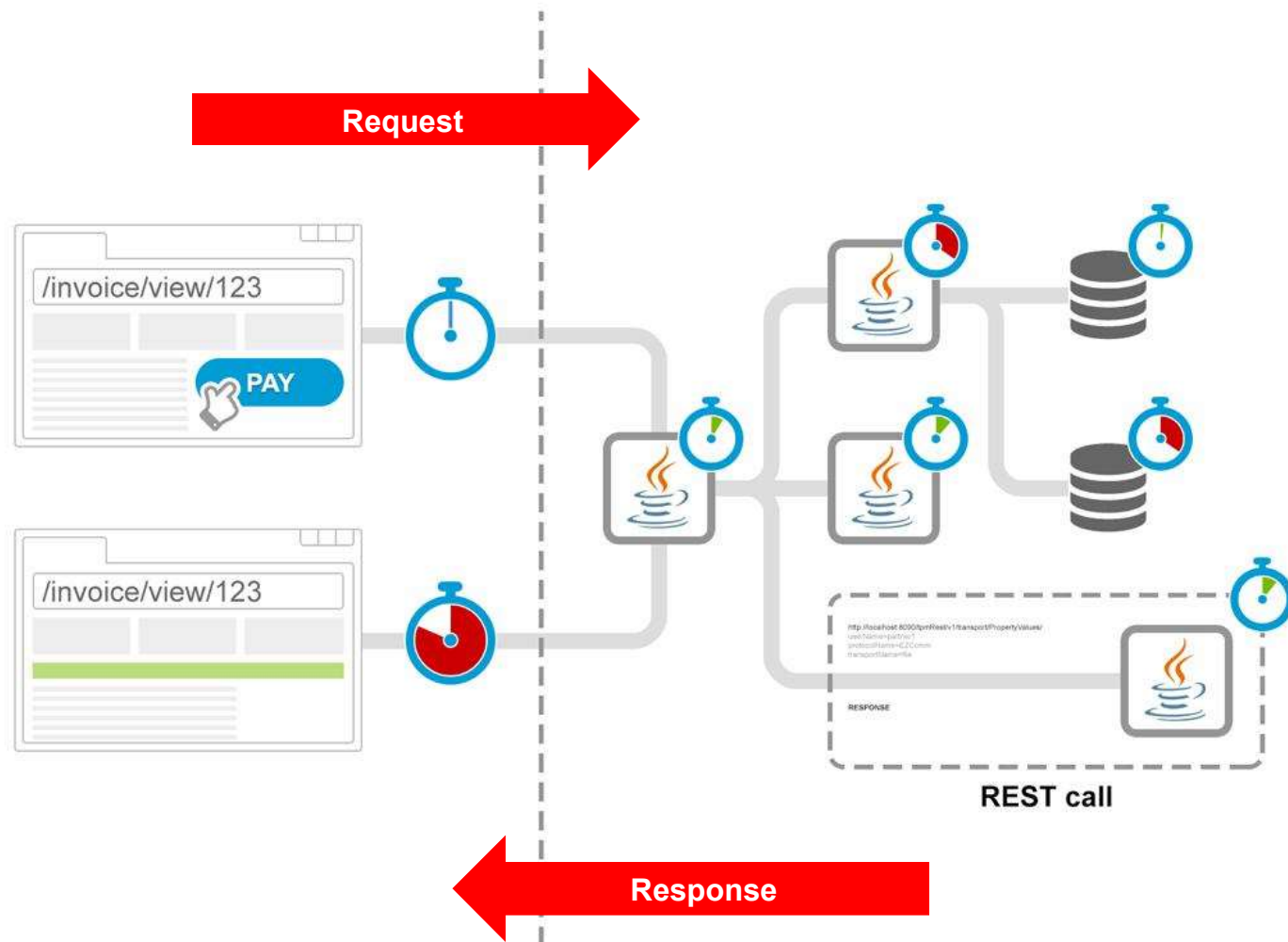
Peter Kassenaar –
info@kassenaar.com

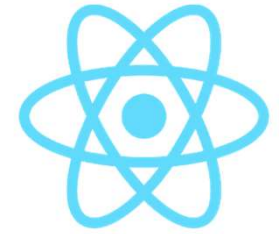


Communicating with external APIs

Fetching and handling external data over http

Calling a backend from your React App

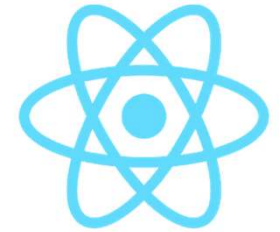




Options for using external data

- Tons of external (open/public) API's available
 - <https://github.com/public-apis/public-apis>
- React can *not* do http-calls by itself.
- Options for calling them from your app
 - `fetch` – natively built in to the browser
 - JQuery `$.ajax()`
 - `axios` – popular open source http-library
 - ...lots of **other options**

Our choice: axios



Neural Processing Mechanisms

npm Enterprise Products Solutions Resources Docs Support

npm Search packages Search

Need private packages and team management tools? [Check out npm Orgs. »](#)

axios

0.19.0 • Public • Published 5 months ago

Readme 2 Dependencies 25.871 Dependents 40 Versions

npm v0.19.0 build failing coverage 94% install size 409 kB downloads 25M/month chat on github code helpers 124

Promise based HTTP client for the browser and node.js

Features

- Make `XMLHttpRequests` from the browser
- Make `http` requests from node.js
- Supports the `Promise` API

install

```
> npm i axios
```

weekly downloads

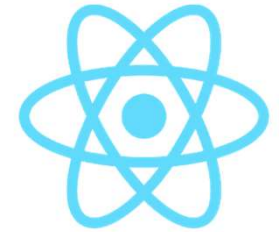
5.921.879

version	license
0.19.0	MIT
open issues	pull requests
583	135

```
npm install axios [--save]
```

<https://www.npmjs.com/package/axios>

API of choice: Restful Countries



REST Countries

View on GitLab

Issues

REST COUNTRIES PE

Donate!

[Changelog](#)

[New in Version 3](#)

[V3 vs V3.1](#)

[Users](#)

API ENDPOINTS V3

All

[Name](#)

[Full Name](#)

[Code](#)

[List of codes](#)

[Currency](#)

[Language](#)

[Capital city](#)

[Region](#)

[Subregion](#)

API ENDPOINTS V2

All

[Name](#)

[Full Name](#)

[Code](#)

[List of codes](#)

[Currency](#)

REST COUNTRIES PE

Get information about countries via a RESTful API

Current version: 3.1

DONATE!

We're getting about 1.6 millions hits each day and that means also bandwidth (25 GB per day!) so costs have obviously increased.
Please, consider making a [donation](#) to help me pay the server's bills, you're welcome to [donate](#) or you can also [buy me a coffee](#).

CHANGELOG

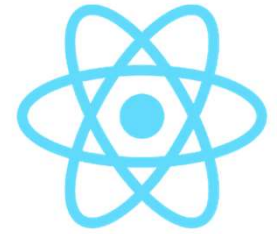
◦ 19/10

◦ Added capital information (lat and long)

◦ Added postal code (format and regex)

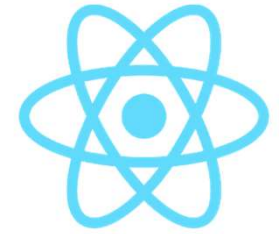
<https://restcountries.com/>

Goal – fetching country data from API



1. **Prepare State** for the data you're about to receive
2. Discover the **URL** that you want to talk to
3. Use `axios` to **call** the url
 - For instance in the `componentDidMount()` lifecycle hook
 - But it can be in a button `clickHandler` as well..
4. **Set the state** in the `.then()` handler of the promise
5. **Pass state** to the component as usual

Set state and call URL



```
import axios from 'axios'  
const url = 'https://restcountries.com/v2/all';
```

URL to talk to

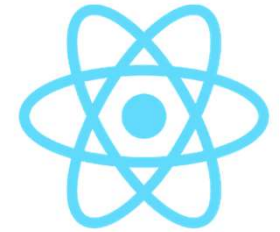
```
state = {  
  error: null,  
  isLoading: false,  
  countries: []  
};
```

Prepare state

```
componentDidMount() {  
  axios.get(url)  
    .then(response => {  
      this.setState({  
        isLoading: true,  
        countries: response.data  
      })  
    })  
}
```

Set state with data

Loading indicator, aka Spinner

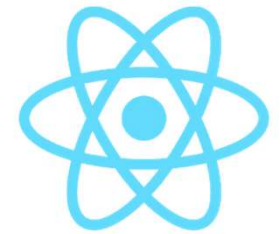


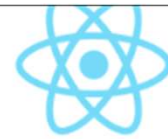
Depends on `this.state.isLoading`. For instance like:

```
{/*Show a Loading indicator as long as the countries are not Loaded*/}
{
  !this.state.isLoading &&
    <LoadingIndicator/>
}
```

```
class LoadingIndicator extends Component {
  render() {
    return (
      <div>
        <img src={require('./spinner.gif')} alt="Please wait..."/>
      </div>
    );
  }
}
```

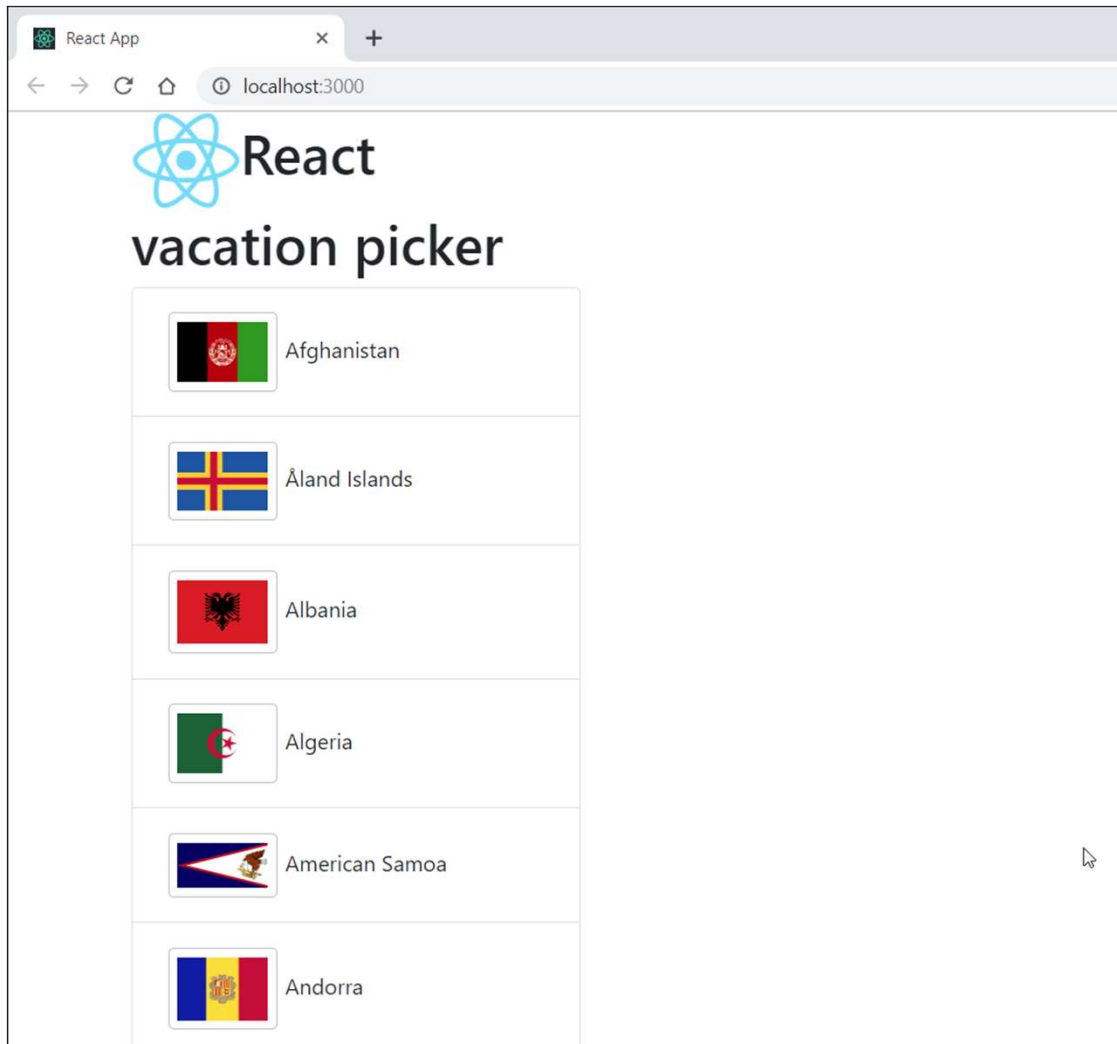
Result w/ simulated delay



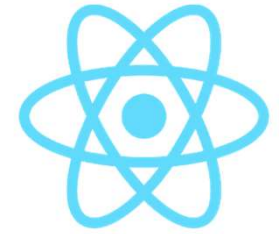
 React
vacation picker



Result after delay



`../examples/500-api-call`



On `async` / `await`

- You *can* use the `async/await` notation if you like.
 - It's just syntactic sugar over Promises

```
async componentDidMount() {  
  const response = await axios.get(url)  
  this.setState({  
    isLoading: true,  
    countries: response.data  
  })  
}
```

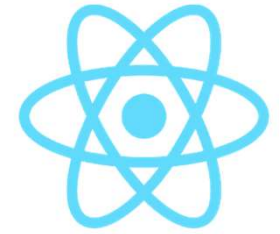


`async / await`



Fetching details

Doing subsequent API calls



Same principle

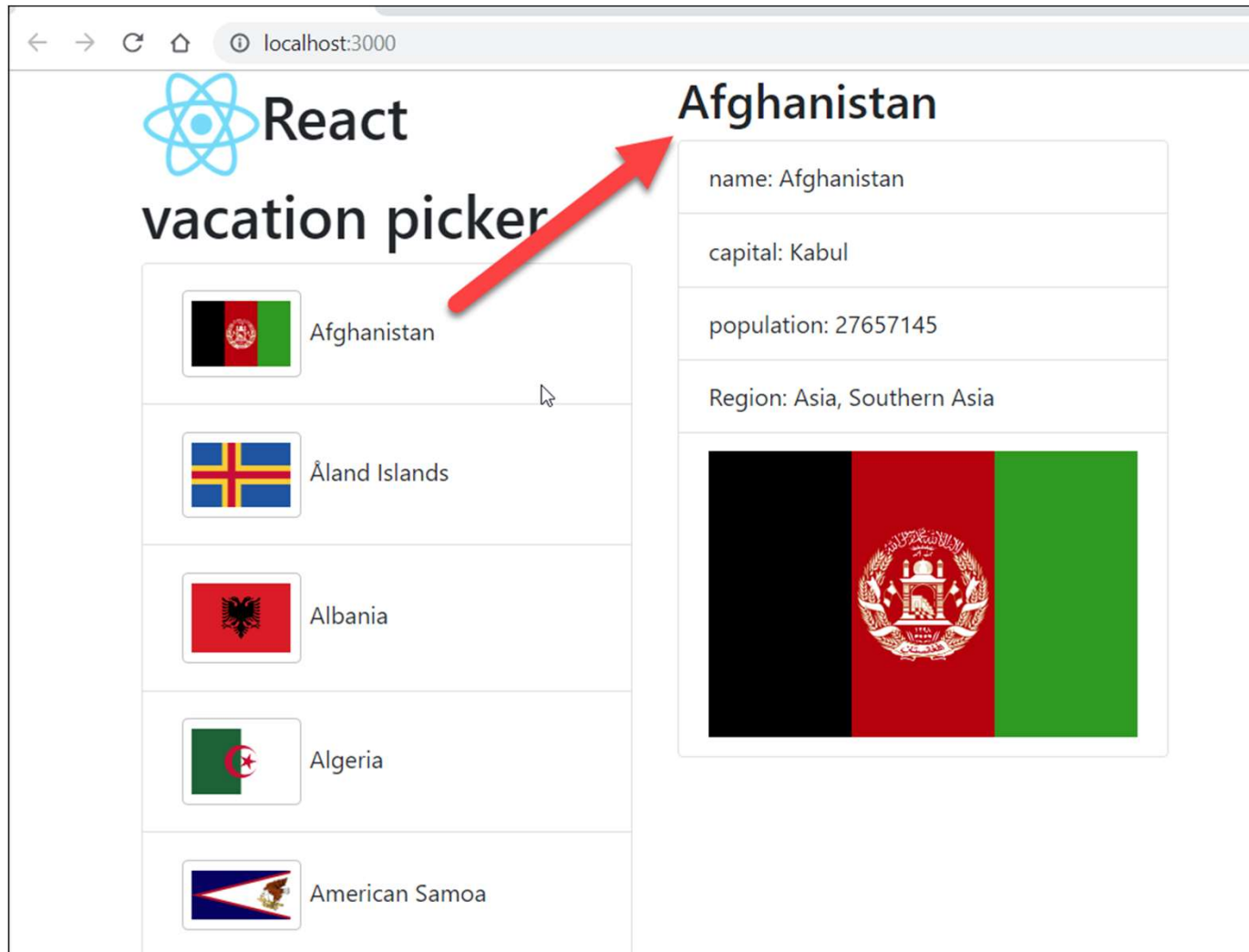
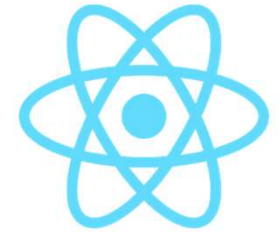
- Pass click handler as prop to `<VacationPicker />`
- find URL, do an `axios` call, update state
- Pass the state to a (new) detail component

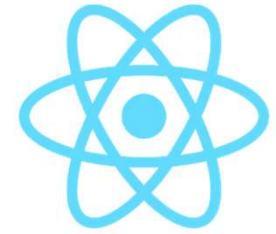
```
<VacationPicker  
  countries={this.state.countries}  
  select={(country) => this.getCountry(country)}  
>
```

Get specific country

```
// get details for a specific country  
getCountry(name) {  
  axios.get(`${detail_url}/${name}`)  
    .then(response => {  
      this.setState({  
        country: response.data[0]  
      })  
    })  
}
```

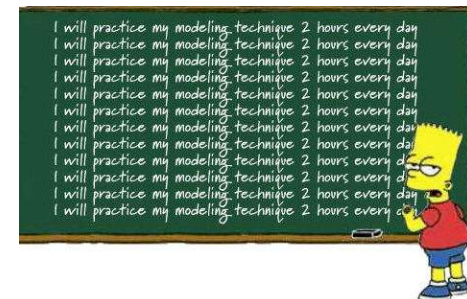
Result



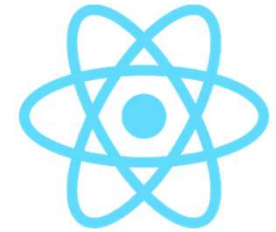


Workshop

- Create a new project, or work from the example project
 - Install `axios` if necessary
- Use the **Typicode API** to fetch dummy random users:
 - <https://jsonplaceholder.typicode.com/users>
- Show usernames and e-mail addresses in a list
- Clicking a user fetches details for that specific user. Show them in a detail component
- Optional – Create a `<SearchCountry>` component for the example app
 - where the user can type in (a part of) the name.
 - Search for that country/countries, show them.
- Example `../500-api-call`



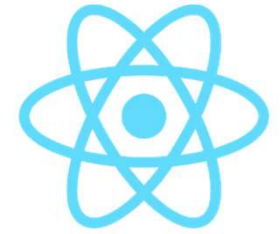
More API's



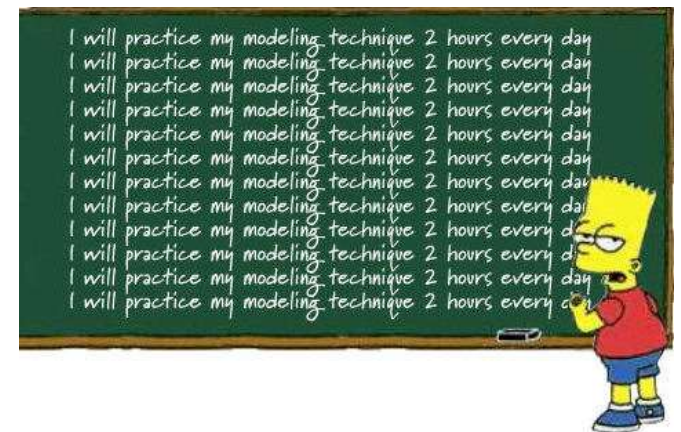
```
VacationPicker.js x JavaScript APIs.txt x
1
2 // DEZE API's zijn open, zonder registratie beschikbaar, of al geregistreerd (reg
3 // THESE API's are open, free, mostly available without registration (or already
4
5 https://opendata.rdw.nl/resource/m9d7-ebf2.json?kenteken= + kenteken
6 http://swapi.co/ - The Star Wars API
7 http://api.openweathermap.org/data/2.5/weather?units=metric&appid=8566d604cd9402b
8 http://ergast.com/mrd/ - Ergast Motor (Formule 1) API
9 https://randomuser.me/api/?results=10 - random user data
10 http://www.omdbapi.com/?apikey=f1f56c8e& - Open Movie Database (gebruik liever je
11 https://api.github.com/users/ - GitHub user information.
12 http://dev.markitondemand.com/Api/v2/Quote/json?symbol=AAPL - Aandelen/stock quot
13 http://restcountries.eu/ - Information on all countries in the world.
14 https://restcountries.eu/rest/v1/all - Andere API met Country-informatie
15 http://api.postcodedata.nl/v1/postcode/?postcode=1211EP&streetnumber=60&ref=domei
16 http://opendata.cbs.nl/dataportaal/portal.html?_la=nl&_catalog=CBS - Honderden AP
17 https://data.pdok.nl/datasets - Allerlei informatie van het Kadaster
18 https://coinmarketcap.com/api/ - Blockchain/Bitcoin information
19 https://www.anwb.nl/feeds/gethf - realtime file-informatie Nederlandse snelwegen
20 https://api.spacexdata.com/v3/launches/ - SpaceX rocket launch information
```

File: JavaScript APIs.txt in the root of the repository

Workshop #2




- Create a small application using one of the other API's in the file `JavaScript API's.txt`
- Store and fetch data in a store. Use for instance:
 - Pokemon API
 - Open Movie Database API
 - OpenWeatherMap API
 - ...





More info



 **React**

Docs Tutorial Blog Community

Q Search

v16.11.0  Languages GitHub 

AJAX and APIs

How can I make an AJAX call?

You can use any AJAX library you like with React. Some popular ones are [Axios](#), [jQuery AJAX](#), and the browser built-in [window.fetch](#).

Where in the component lifecycle should I make an AJAX call?

You should populate data with AJAX calls in the `componentDidMount` lifecycle method. This is so you can use `setState` to update your component when the data is retrieved.

Example: Using AJAX results to set local state

The component below demonstrates how to make an AJAX call in `componentDidMount` to populate local component state.

INSTALLATION ▾

MAIN CONCEPTS ▾

ADVANCED GUIDES ▾

API REFERENCE ▾

HOOKS ▾

TESTING ▾

CONCURRENT MODE (EXPERIMENTAL) ▾

CONTRIBUTING ▾

FAQ ^

AJAX and APIs

Babel, JSX, and Build Steps

Passing Functions to Components

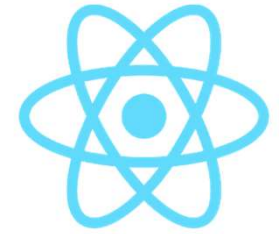
Component State

Styling and CSS

File Structure

<https://reactjs.org/docs/faq-ajax.html>

Checkpoint



- You know how to make Ajax/[API calls](#) from an application
- You are able to [handle data](#) coming from a Restful endpoint and show it in your app
- ALWAYS read the [documentation](#) of the corresponding API