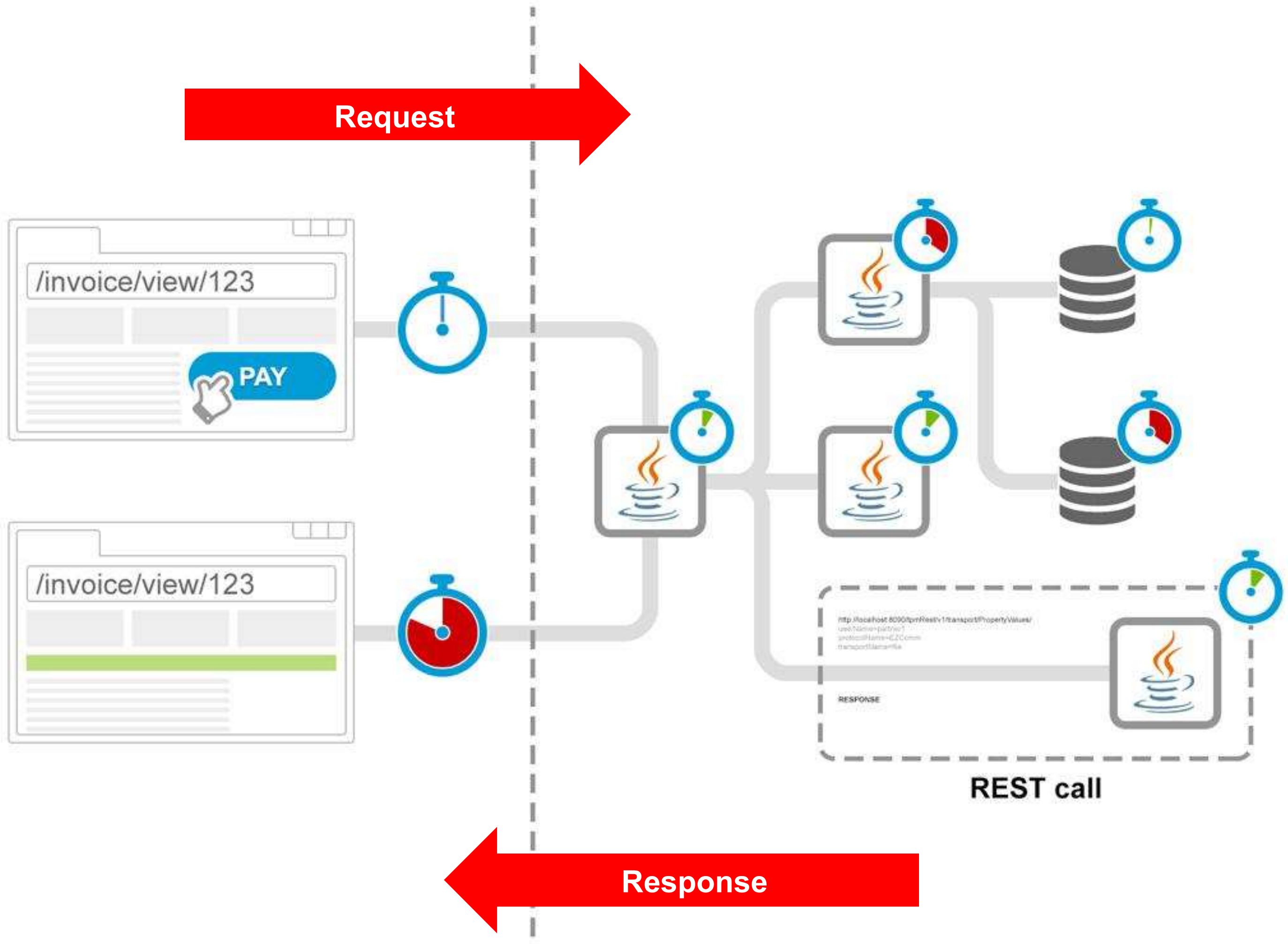
React Fundamentals
Module – Http communication

Peter Kassenaar –
info@kassenaar.com

Communicating with external APIs

Fetching and handling external data over http

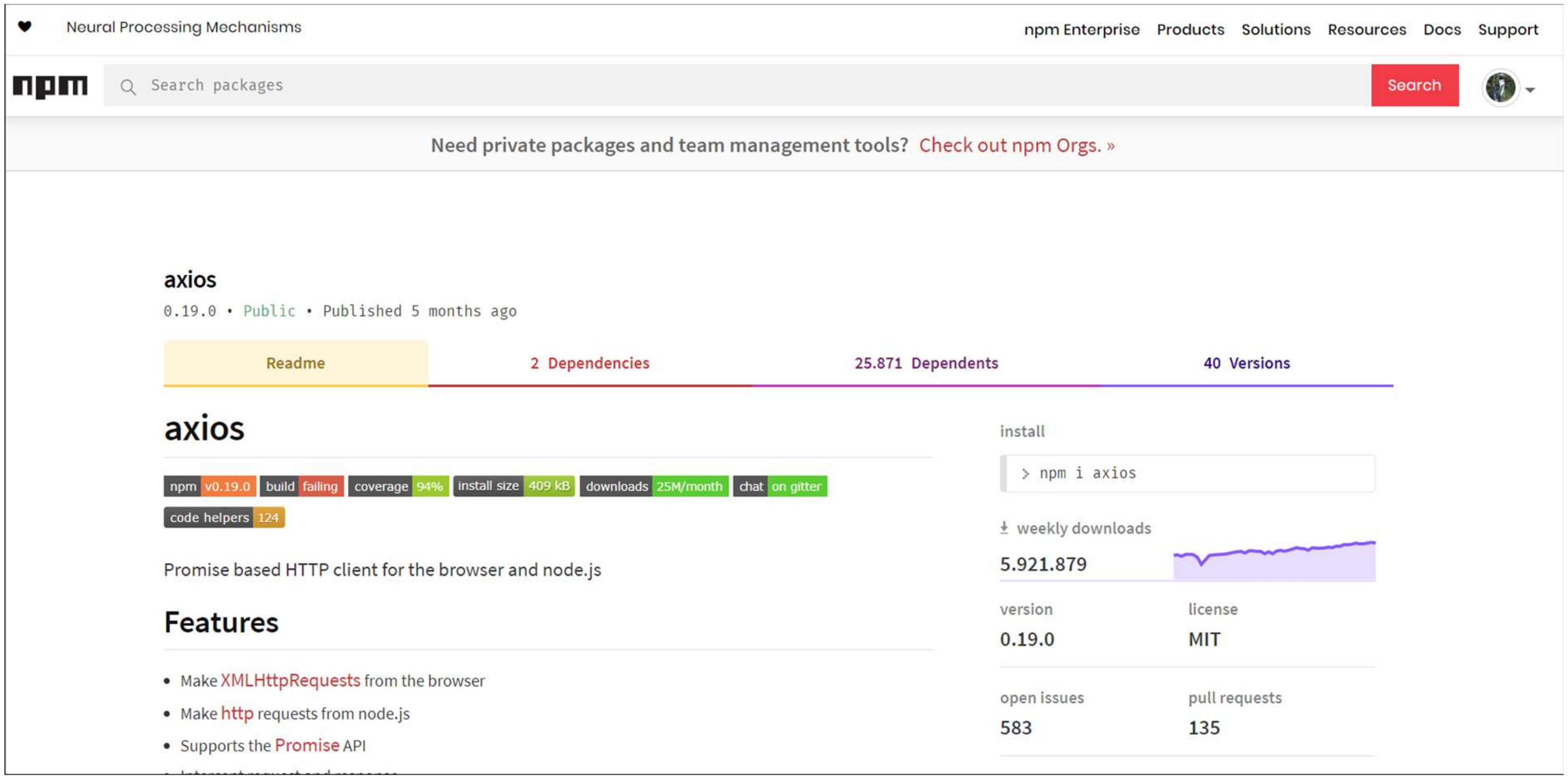
Calling a backend from your React App



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

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

axios

npm v0.19.0 | build failing | coverage 94% | install size 409 kB | downloads 25M/month | chat on gitter | 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

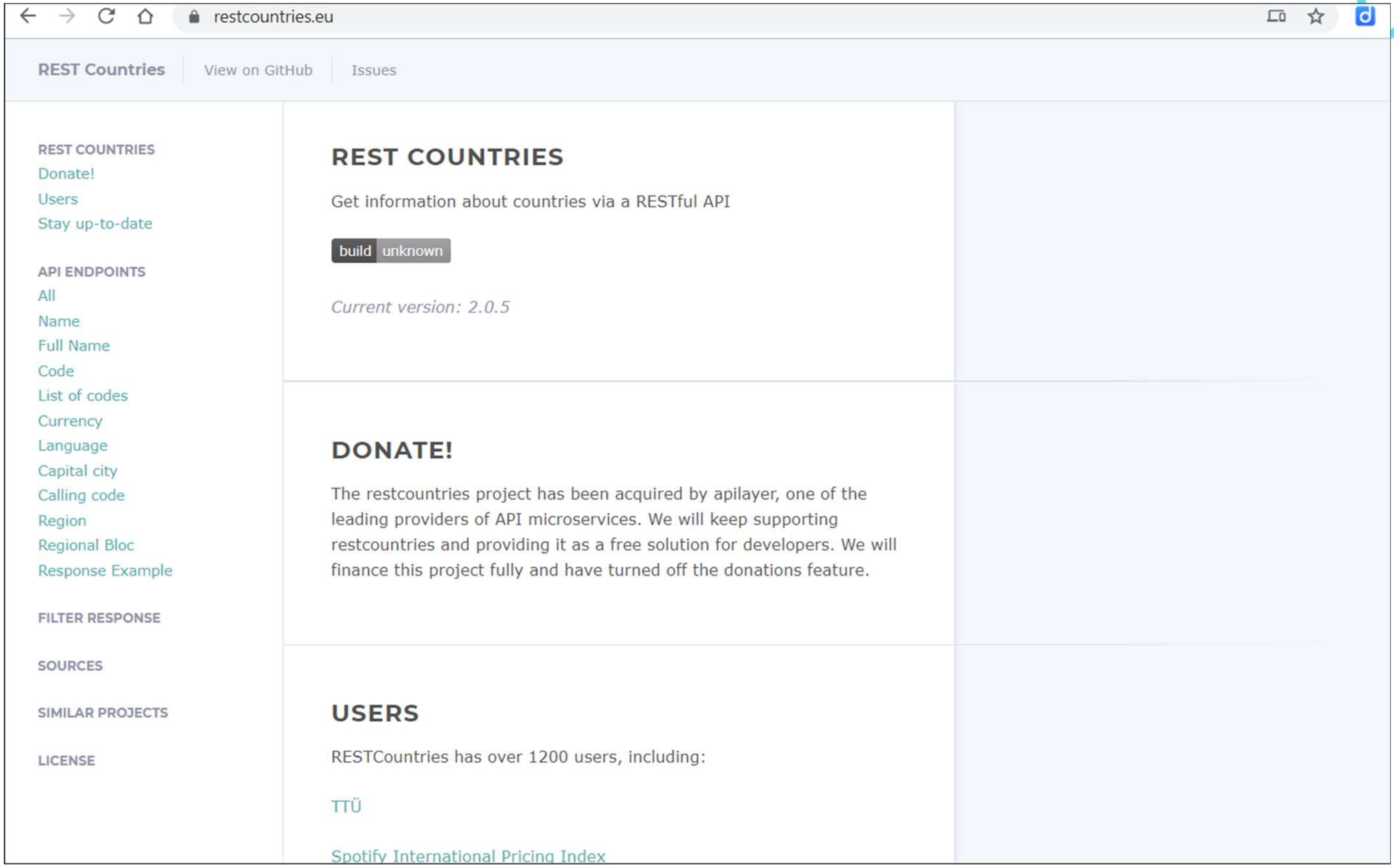
version	license
0.19.0	MIT

open issues	pull requests
583	135

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

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

API of choice: Restful Countries



restcountries.eu

REST Countries | View on GitHub | Issues

REST COUNTRIES
Donate!
Users
Stay up-to-date

API ENDPOINTS
All
Name
Full Name
Code
List of codes
Currency
Language
Capital city
Calling code
Region
Regional Bloc
Response Example

FILTER RESPONSE

SOURCES

SIMILAR PROJECTS

LICENSE

REST COUNTRIES

Get information about countries via a RESTful API

build unknown

Current version: 2.0.5

DONATE!

The restcountries project has been acquired by apilayer, one of the leading providers of API microservices. We will keep supporting restcountries and providing it as a free solution for developers. We will finance this project fully and have turned off the donations feature.

USERS

RESTCountries has over 1200 users, including:

TTÜ

Spotify International Pricing Index

https://restcountries.eu/

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.eu/rest/v2/all';
```

URL to talk to

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

Prepare state

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

Set state with data

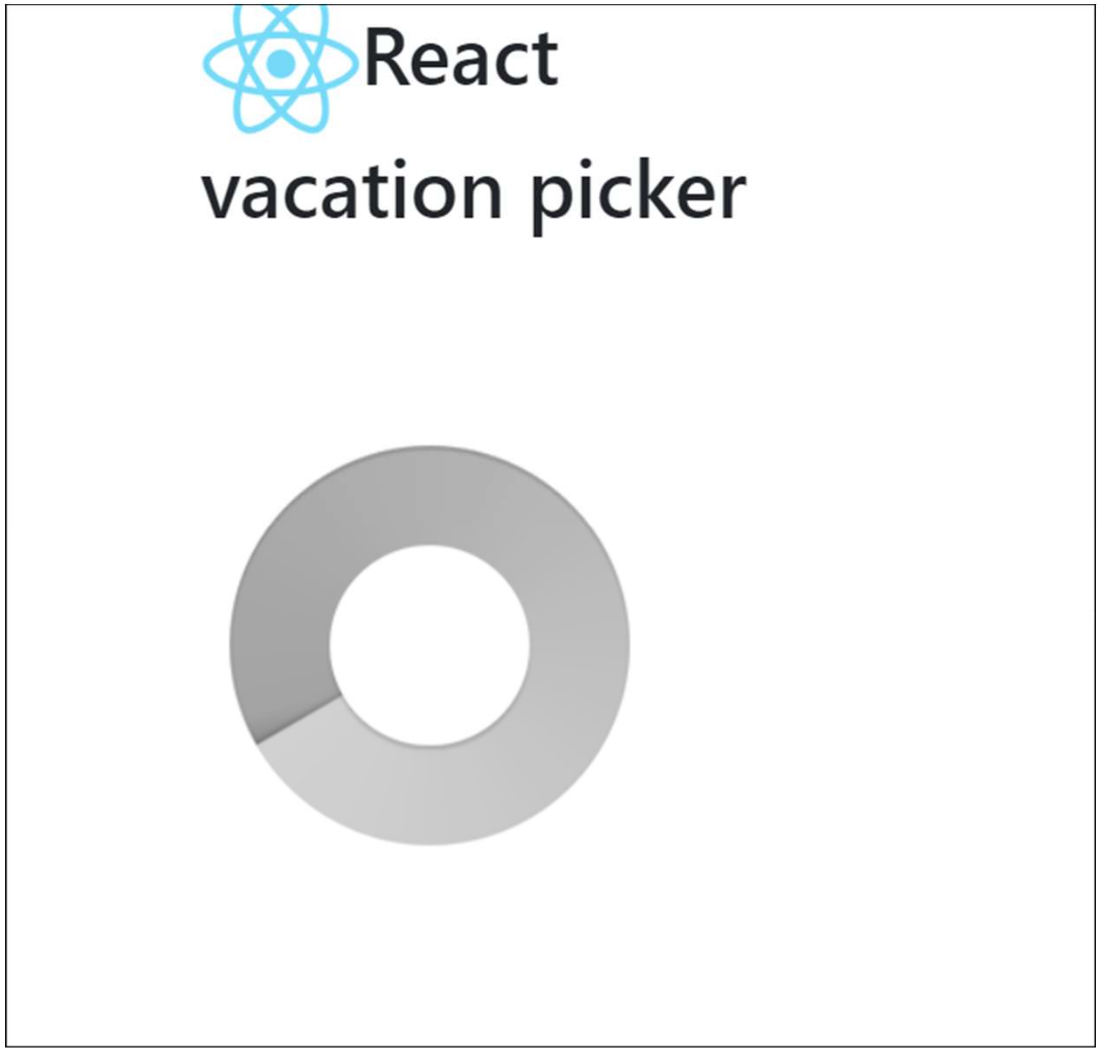
Loading indicator, aka Spinner

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

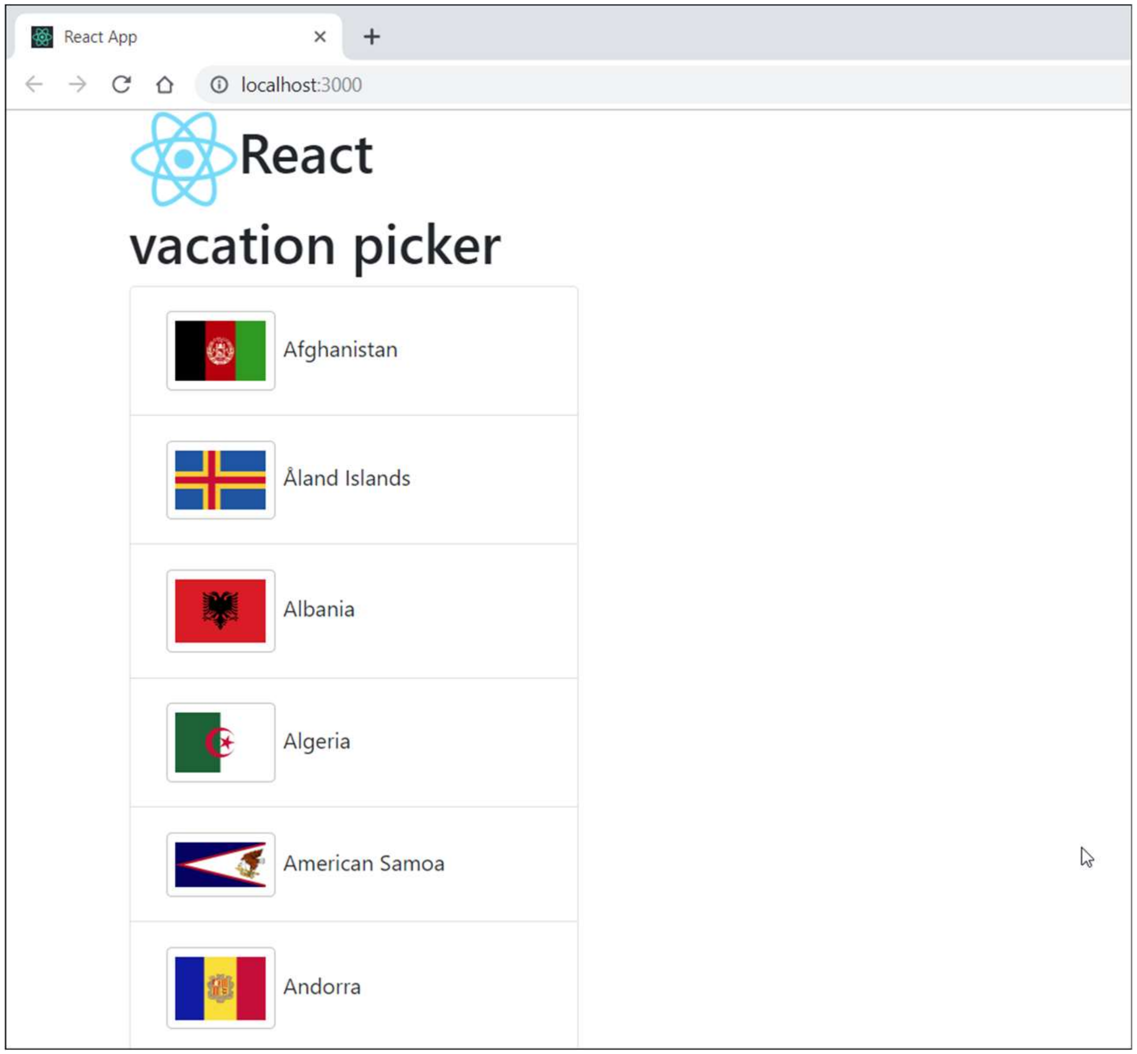
```
{/*Show a loading indicator as long as the countries are not loaded*/}
{
    !this.state.isLoaded &&
       <LoadingIndicator/>
}
```

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

Result w/ simulated delay



Result after delay


React App

localhost:3000

React

vacation picker

Afghanistan

Åland Islands

Albania

Algeria

American Samoa

Andorra

../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({
        isLoaded: true,
        countries: response.data
    })
}
```

async / await

Fetching details

Doing subsequent API calls

Same principle

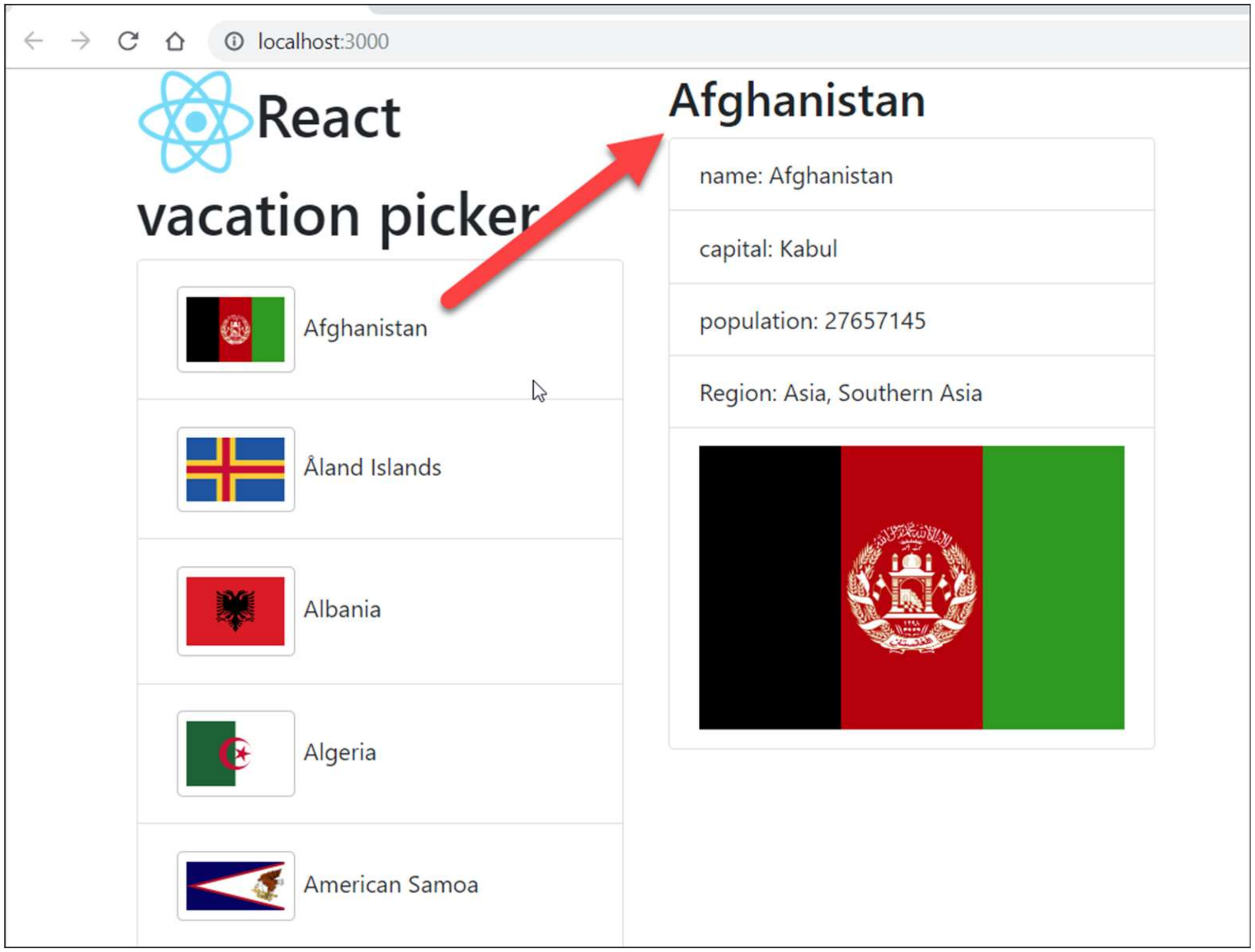
- Pass click handler as prop to `<VacationPicker />`
- find URL, do an `axios` call, update state
- Pass the state to a (new) detail component

```
<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



React

vacation picker

Afghanistan

Åland Islands

Albania

Algeria

American Samoa

Afghanistan

name: Afghanistan

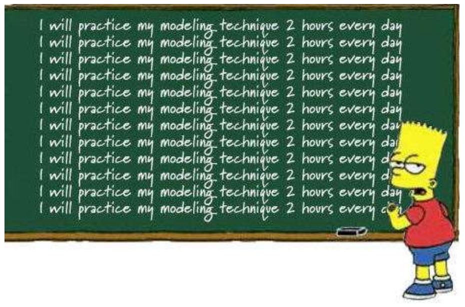
capital: Kabul

population: 27657145

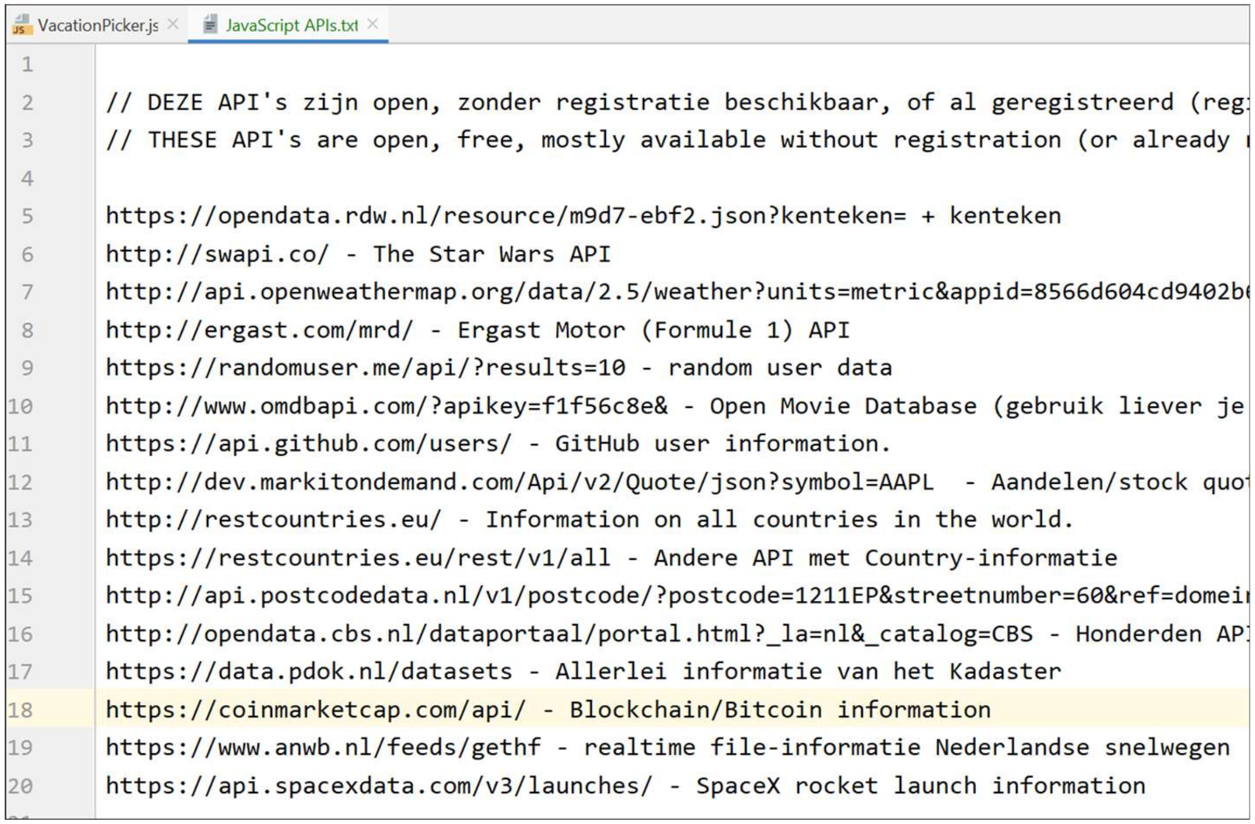
Region: Asia, Southern Asia

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

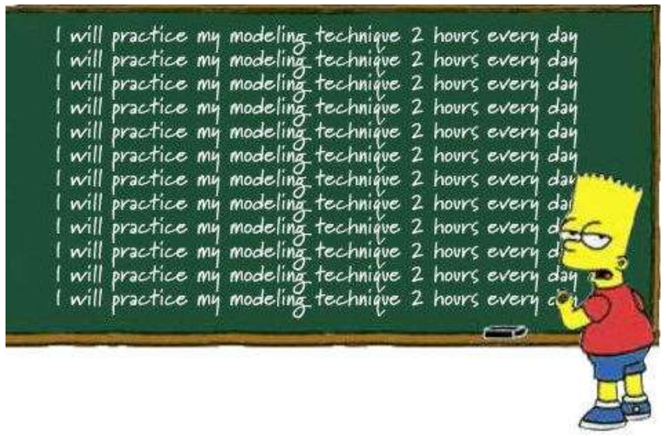


```

// DEZE API's zijn open, zonder registratie beschikbaar, of al geregistreerd (reg
// THESE API's are open, free, mostly available without registration (or already

https://opendata.rdw.nl/resource/m9d7-ebf2.json?kenteken= + kenteken
http://swapi.co/ - The Star Wars API
http://api.openweathermap.org/data/2.5/weather?units=metric&appid=8566d604cd9402b
http://ergast.com/mrd/ - Ergast Motor (Formule 1) API
https://randomuser.me/api/?results=10 - random user data
http://www.omdbapi.com/?apikey=f1f56c8e& - Open Movie Database (gebruik liever je
https://api.github.com/users/ - GitHub user information.
http://dev.markitondemand.com/Api/v2/Quote/json?symbol=AAPL  - Aandelen/stock quo
http://restcountries.eu/ - Information on all countries in the world.
https://restcountries.eu/rest/v1/all - Andere API met Country-informatie
http://api.postcodedata.nl/v1/postcode/?postcode=1211EP&streetnumber=60&ref=domei
http://opendata.cbs.nl/dataportaal/portal.html?_la=nl&_catalog=CBS - Honderden AP
https://data.pdok.nl/datasets - Allerlei informatie van het Kadaster
https://coinmarketcap.com/api/ - Blockchain/Bitcoin information
https://www.anwb.nl/feeds/gethf - realtime file-informatie Nederlandse snelwegen
https://api.spacexdata.com/v3/launches/ - SpaceX rocket launch information
```

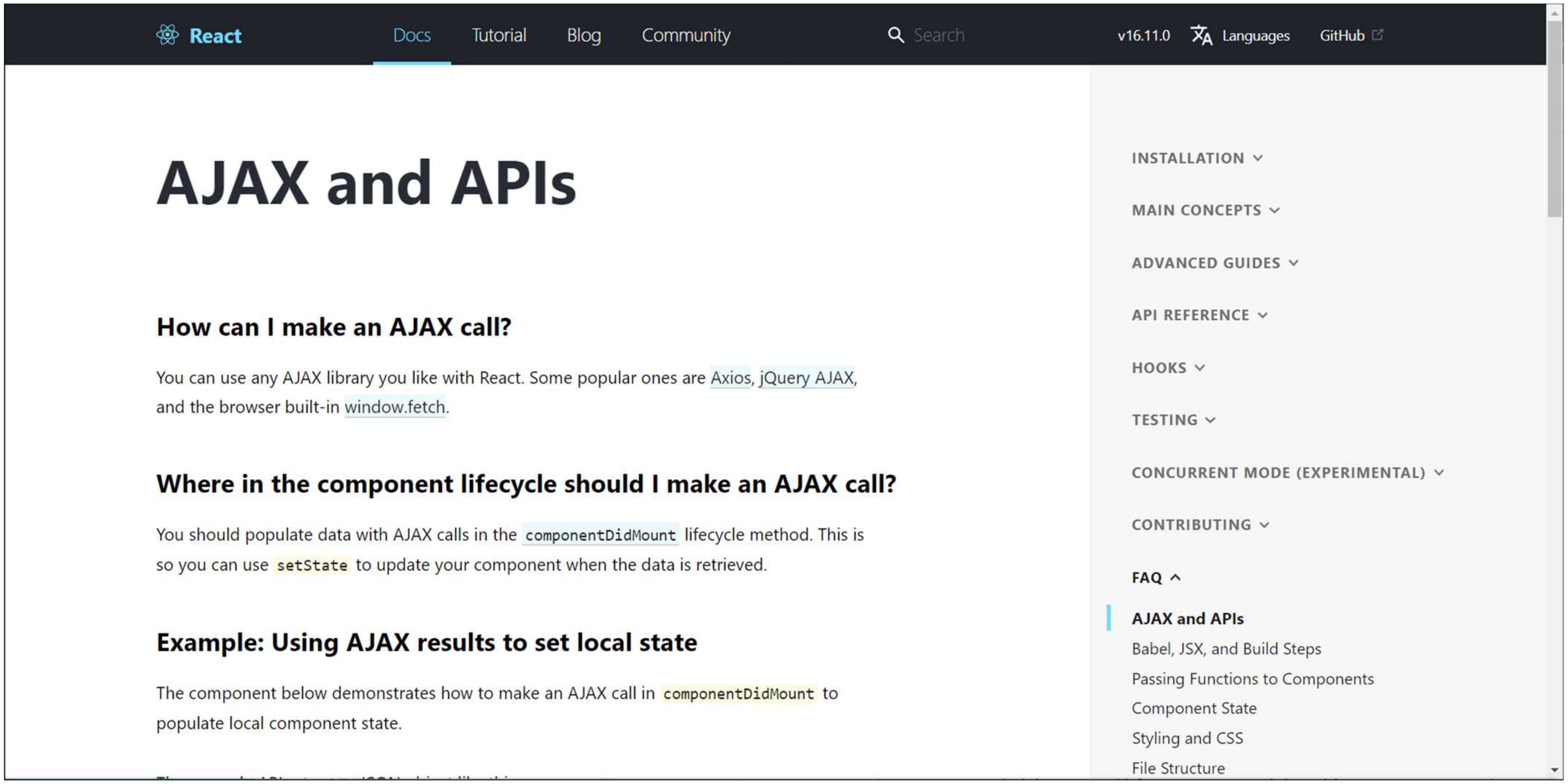
File: `JavaScript APIs.txt` in the root of the repository

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



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.

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