



Vue Fundamentals

Using Http and axios

Peter Kassenaar –
info@kassenaar.com

Peter Kassenaar

INCLUSIEF
GRATIS
WEBVERSIE
VAN HET
BOEK

Vue.js

Web Development Library

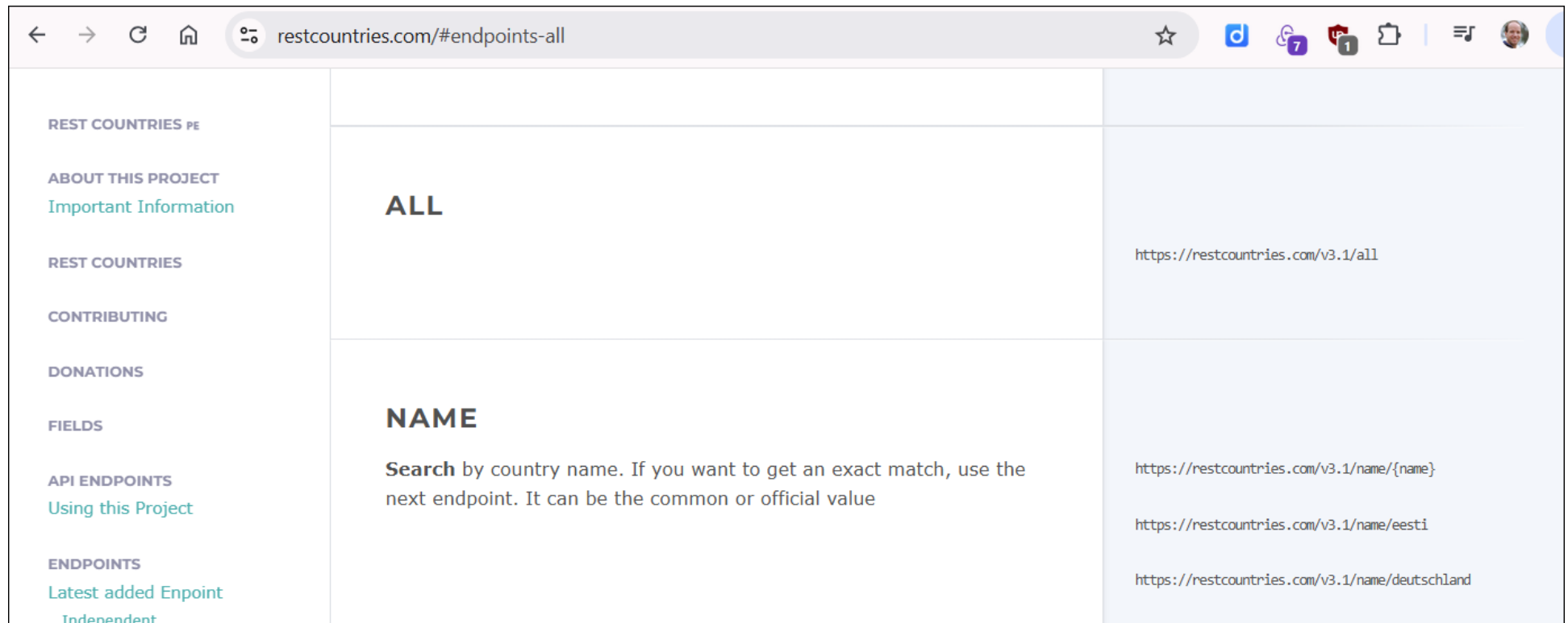


VANDUUREN MEDIA

P. 221 e.v.

Consuming data from an API

- We're going to talk to a 'real' country-API on the internet
 - <https://restcountries.com/>
- Using `http` is *not* built into `Vue.js`

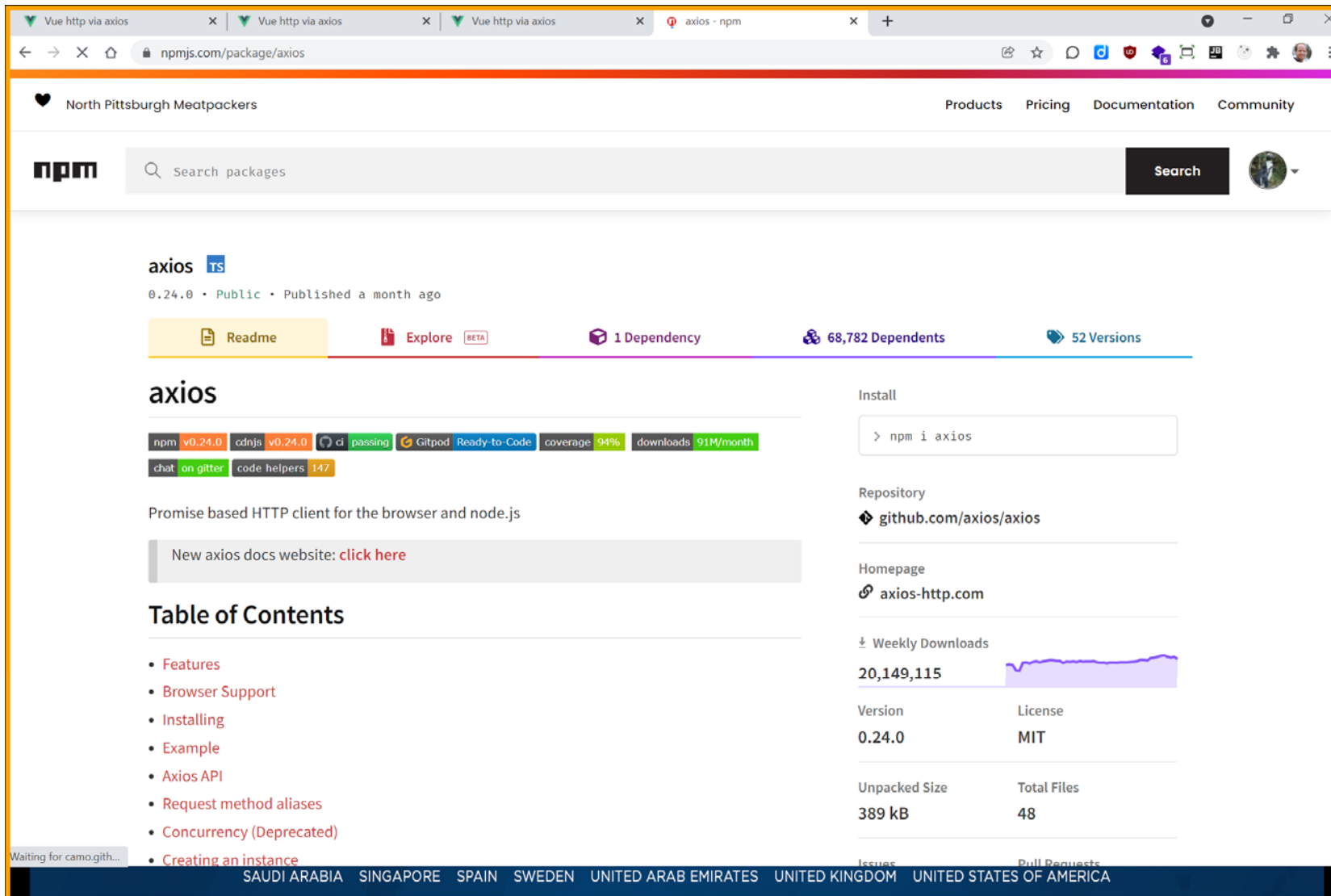


Using Axios

*“There are many times when building application for the web that you may want to consume and display data from an API. There are several ways to do so, but a very popular approach is to use **axios**, a promise-based HTTP client.”*

<https://github.com/axios/axios>

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



The screenshot shows the npm package page for axios. The browser's address bar displays 'npmjs.com/package/axios'. The page header includes the npm logo, a search bar, and navigation links for Products, Pricing, Documentation, and Community. The main content area features the package name 'axios' with a TypeScript (TS) tag, version '0.24.0', and publication status 'Public'. It also shows '1 Dependency', '68,782 Dependents', and '52 Versions'. A horizontal bar contains links to Readme, Explore, and other resources. Below this, the package description 'Promise based HTTP client for the browser and node.js' is visible, along with a link to the new docs website. A 'Table of Contents' section lists various features and installation instructions. On the right side, there is an 'Install' section with a terminal command, a 'Repository' link to the GitHub repository, a 'Homepage' link, and a 'Weekly Downloads' chart showing 20,149,115 downloads. A table at the bottom right lists package details: Version (0.24.0), License (MIT), Unpacked Size (389 kB), and Total Files (48). The footer of the page lists various countries: SAUDI ARABIA, SINGAPORE, SPAIN, SWEDEN, UNITED ARAB EMIRATES, UNITED KINGDOM, and UNITED STATES OF AMERICA.

North Pittsburgh Meatpackers

Products Pricing Documentation Community

npm Search packages Search

axios **TS**

0.24.0 • Public • Published a month ago

Readme Explore **BETA** 1 Dependency 68,782 Dependents 52 Versions

axios

npm v0.24.0 cdnjs v0.24.0 ci passing Gitpod Ready-to-Code coverage 94% downloads 91M/month

chat on gitter code helpers 147

Promise based HTTP client for the browser and node.js

New axios docs website: [click here](#)

Table of Contents

- Features
- Browser Support
- Installing
- Example
- Axios API
- Request method aliases
- Concurrency (Deprecated)
- Creating an instance

Install

```
> npm i axios
```

Repository

[github.com/axios/axios](#)

Homepage

[axios-http.com](#)

Weekly Downloads

20,149,115

Version	License
0.24.0	MIT

Unpacked Size	Total Files
389 kB	48

Issues Pull Requests

SAUDI ARABIA SINGAPORE SPAIN SWEDEN UNITED ARAB EMIRATES UNITED KINGDOM UNITED STATES OF AMERICA

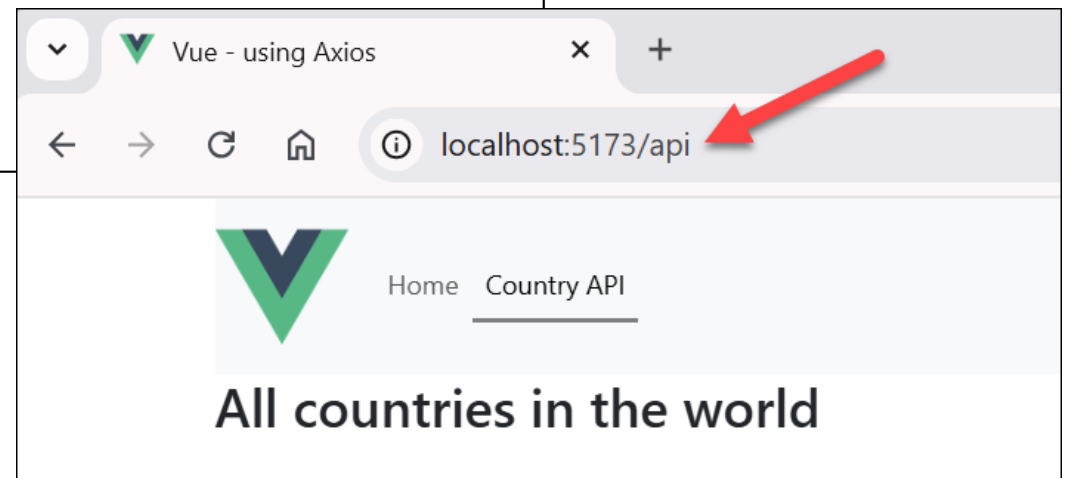
Installing axios

- Add axios to your project
 - `npm install axios`
- **Import axios** in the component where you want to use it.
- It is actually considered *bad* practice to use axios or Http-calls directly from a component!
 - Recommended – use a store mechanism, like Vuex or Pinia.
 - We're going to cover that later on.
- We're doing it here to **show the basics**

New route

- We're creating a new route and a new component to navigate.
 - Create `ApiComponent.vue`
 - Just an empty component for now
 - Add the route like so:

```
// Component for using to the country API RestCountries  
{  
  path: '/api',  
  name: 'api',  
  component: ApiComponent  
}
```



Using axios

- Call axios inside the component
- Assign results to reactive variables

```
import axios from 'axios';
import {ref} from "vue";

const url = 'https://restcountries.com/v3.1/all?fields=.....'
const countries = ref([]);
const errors = ref([]);

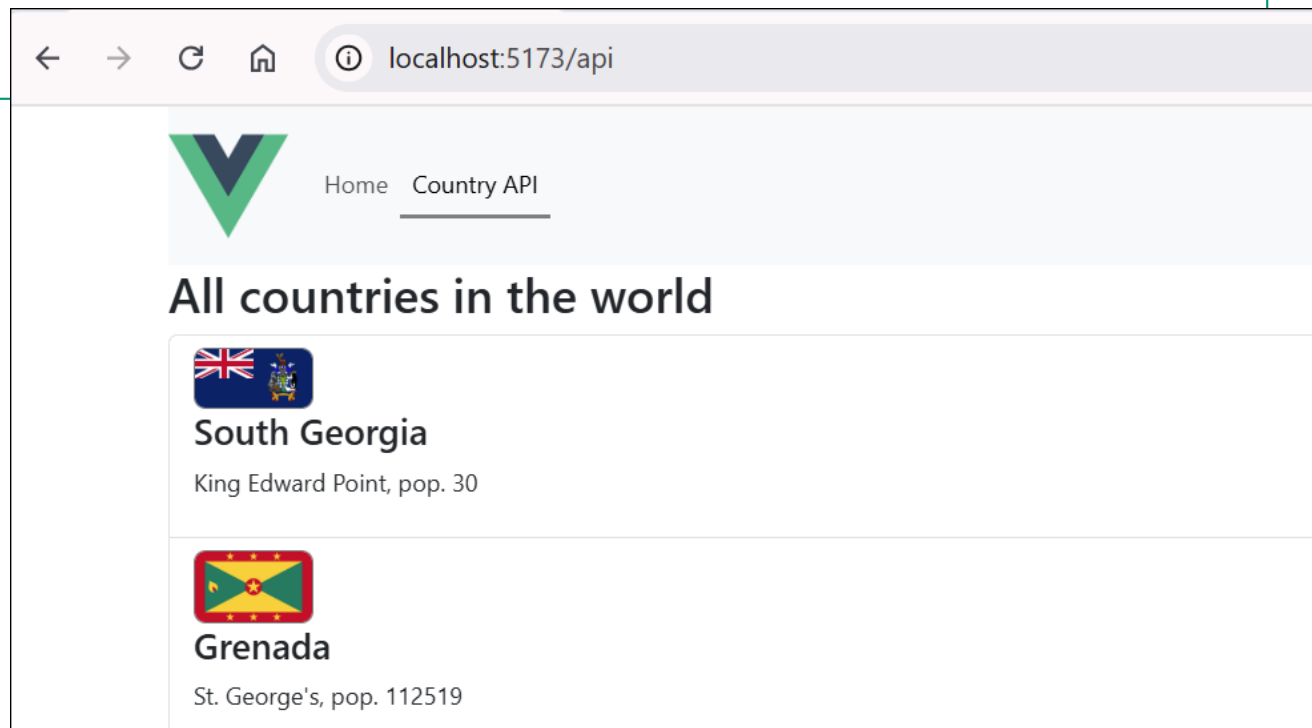
axios.get(url)
  .then(response => countries.value = response.data)
  .catch(error => errors.value.push(error));
```


Update your HTML

```
<ul class="list-group" v-if="countries && countries.length">
  <li class="list-group-item" v-for="country of countries" :key="country.name">
    <span class="float-left">
      
    </span>
    <h4>{{ country.name.common }} </h4>
    <p>{{ country.capital ? country.capital[0] : '' }},
      pop. {{ country.population }}
    </p>
  </li>
</ul>
```

Of course, inspect the structure of the data you get in your response and update the HTML accordingly

Result



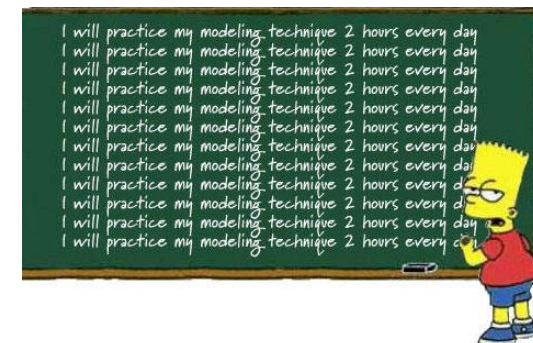
Async/await

- You can use **async/await** with axios
 - Not supported in older browsers (IE 11)
 - Results in the UI are identical

```
// async/await version - results in UI are identical  
// but we have to use the onMounted lifecycle hook in this case  
onMounted(async () =>{  
  try {  
    const response = await axios.get(url)  
    countries.value = response.data;  
  } catch (error) {  
    errors.value.push(errors);  
  }  
})
```

Workshop

- Use your own application, or start from [../330-http-axios](#)
- Add a new component <ApiCountryDetail>
- If a country is selected, go to a country detail page for that country.
- Fetch info for that country over `axios`, show detail component
 - Tip : use the name of `fullname` routes of the API. See <https://restcountries.com/#endpoints-name> for details
- Optional – Create an <ApiSearchCountry> component, where the user can type in (a part of) the name.
 - Search for that country/countries, show them.
 - When clicked, it should also load the detail component
 - Possible Solution ([../335-workshop](#))



Checkpoint

- You can perform **http-calls** from your Vue application using axios
- You are able to handle the results from an **external API** and show them in the UI.
- You can do **promise-based** calls and **async/await**-based calls