*Vue.js REST API Consumption with Axios*

## Installation

Axios should be installed from NPM or Yarn:

# Yarn

$ yarn add axios

# NPM

$ npm install axios --save

## Populating Data with a GET Request

While it’s generally poor practice, you can use Axios directly in your components to fetch data from a method, lifecycle hook, or whenever.

ExampleComponent.vue

<template>

<ul v-if="posts && posts.length">

<li v-for="post of posts">

<p><strong>{{post.title}}</strong></p>

<p>{{post.body}}</p>

</li>

</ul>

<ul v-if="errors && errors.length">

<li v-for="error of errors">

{{error.message}}

</li>

</ul>

</template>

<script>

import axios from 'axios';

export default {

data() {

return {

posts: [],

errors: []

}

},

// Fetches posts when the component is created.

created() {

axios.get(`http://jsonplaceholder.typicode.com/posts`)

.then(response => {

// JSON responses are automatically parsed.

this.posts = response.data

})

.catch(e => {

this.errors.push(e)

})

// async / await version (created() becomes async created())

//

// try {

// const response = await axios.get(`http://jsonplaceholder.typicode.com/posts`)

// this.posts = response.data

// } catch (e) {

// this.errors.push(e)

// }

}

}

</script>

## ushing Data with a POST Request

One can send POST, PUT, PATCH, and DELETE requests just as easily.

(Note, it’s a terrible idea to send requests on every input event, at the very least do some throttling or debouncing.)

ExampleComponent.vue

<template>

<input type="text" v-model="postBody" @change="postPost()"/>

<ul v-if="errors && errors.length">

<li v-for="error of errors">

{{error.message}}

</li>

</ul>

</template>

<script>

import axios from 'axios';

export default {

data() {

return {

postBody: '',

errors: []

}

},

// Pushes posts to the server when called.

postPost() {

axios.post(`http://jsonplaceholder.typicode.com/posts`, {

body: this.postBody

})

.then(response => {})

.catch(e => {

this.errors.push(e)

})

// async / await version (postPost() becomes async postPost())

//

// try {

// await axios.post(`http://jsonplaceholder.typicode.com/posts`, {

// body: this.postBody

// })

// } catch (e) {

// this.errors.push(e)

// }

}

}

</script>

## A Common Base Instance

A frequently overlooked but very useful capability Axios provides is the ability to create a base instance that allows you to share a common base URL and configuration across all calls to the instance. This comes in handy if all of your calls are to a particular server, or need to share headers, such as an authorization header.

http-common.js

import axios from 'axios';

export const HTTP = axios.create({

baseURL: `http://jsonplaceholder.typicode.com/`,

headers: {

Authorization: 'Bearer {token}'

}

})

You could now use HTTP like so,

<script>

import {HTTP} from './http-common';

export default {

data() {

return {

posts: [],

errors: []

}

},

created() {

HTTP.get(`posts`)

.then(response => {

this.posts = response.data

})

.catch(e => {

this.errors.push(e)

})

}

}

</script>