Web Application Development

Produced by

David Drohan (ddrohan@wit.ie)

Department of Computing & Mathematics
Waterford Institute of Technology

http://www.wit.ie

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Vue.js

RELATED

USING AXIOS TO CONSUME DONATION-SERVER API

Axios

A PROMISE-BASED HTTP CLIENT

Vue.js

What is Axios?

- Promise based HTTP client for the browser and node.js
- Features include
  - Make XMLHttpRequests from the browser
  - Make http requests from node.js
  - Supports the Promise API
  - Intercept request and response
  - Transform request and response data
  - Cancel requests
  - Automatic transforms for JSON data
  - Client side support for protecting against XSRF (Cross-Site Request Forgery, aka one-click attack)
- Supported by all Major Browsers

https://github.com/axios/axios

Installing

Using npm:

```
$ npm install axios
```

Using bower:

```
$ bower install axios
```

Using cdn:

```
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
```

DonationVue & Axios

There are many times when building applications for the web that you may want to consume and display data from an API. Our DonationVue Web App is no exception. There are several ways to do this, but a very popular approach is to use ***axios***, which as already mentioned, is a promise-based HTTP client.

We've already built the Server and Api (hosted at https://donationweb-server.herokuapp.com) which is a big help ☺ but that isn't always the case – most of the time all you know about the server (with your developer hat on) is the 'endpoints' to 'hit' when requesting and/or sending data.

This section will take a quick look at how we interact with our Heroku node web server through our Vue Web App and Axios.

DonationVue ‘Services’ Setup

donationservice.js

api.js

.../src/services/api.js [donationvue-3.0]

donationvue-3.0 > src > services > api.js

api.js

```
1  import axios from 'axios'
2
3  export default() => {
4    return axios.create({
5      // baseURL: 'http://localhost:3000/'
6      baseURL: 'https://donationweb-server.herokuapp.com/'
7    })
8  }
```

<anonymous>()

6:57 LF UTF-8 Git: master

.../src/services/donationservice.js [donationvue...

donationservice.js

```
1   import Api from '@/services/api'
2
3   export default {
4     fetchDonations () {...},
7     postDonation (donation) {...},
11    upvoteDonation (id) {...},
14    deleteDonation (id) {...},
17    fetchDonation (id) {...},
20    putDonation (id, donation) {...}
26  }
27
```

16:4 LF UTF-8 Git: master

DonationWeb Revisited

Let’s remind ourselves of the Api we created (back in the day!)

Resource	URI (structure)	HTTP Request
List of Donations	/donations	GET
Get a Single Donation	/donations/{id}	GET
Upvote a Donation	/donations/{id}/vote	PUT
Delete a Donation	/donations/{id}	DELETE
Update a Donation	/donations/{id}	PUT
Add a Donation	/donations/{id}	POST
Total of Donation Votes	/donations/votes	GET

DonationVue ‘Mapped’

Here’s our client side methods mapped to the server side endpoints

Resource	URI (structure)	HTTP Request
`fetchDonations()`	/donations	GET
`fetchDonation(…)`	/donations/{id}	GET
`upvoteDonation(…)`	/donations/{id}/vote	PUT
`deleteDonation(…)`	/donations/{id}	DELETE
`putDonation(…)`	/donations/{id}	PUT
`postDonation(…)`	/donations/{id}	POST
Total of Donation Votes	/donations/votes	GET

fetchDonations()

```
fetchDonations () {
  return Api().get('/donations')
},
```

```
DonationService.fetchDonations()
  .then(response => {
    // JSON responses are automatically parsed.
    this.donations = response.data
    console.log(this.donations)
  })
  .catch(error => {
    this.errors.push(error)
    console.log(error)
  })
```

id passed from Donations.vue

fetchDonation()

```
fetchDonation (id) {
  return Api().get(`/donations/${id}`)
},
```

```
DonationService.fetchDonation(this.$router.params)
  .then(response => {
    this.temp = response.data
    this.donation = this.temp[0]
    this.childDataLoaded = true
    console.log('Getting Donation in Edit: ' +
      JSON.stringify(this.donation, null, 5))
  })
  .catch(error => {
    this.errors.push(error)
    console.log(error)
  })
```

upvoteDonation()

```
upvoteDonation (id) {
  return Api().put(`/donations/${id}/vote`)
},
```

```
DonationService.upvoteDonation(id)
  .then(response => {
    // JSON responses are automatically parsed.
    this.loadDonations()
    console.log(response)
  })
  .catch(error => {
    this.errors.push(error)
    console.log(error)
  })
```

getting our list again

deleteDonation()

```
deleteDonation (id) {
  return Api().delete(`/donations/${id}`)
},
```

```
DonationService.deleteDonation(id)
  .then(response => {
    // JSON responses are automatically parsed.
    this.message = response.data
    console.log(this.message)
    this.loadDonations()
    // Vue.nextTick(() => this.$refs.vuetable.refresh())
    this.$swal('Deleted', 'You successfully deleted this Donation ' +
      JSON.stringify(response.data, null, 5), 'success')
  })
  .catch(error => {
    this.$swal('ERROR', 'Something went wrong trying to Delete ' +
      error, 'error')
    this.errors.push(error)
    console.log(error)
  })
```

sweetAlerts

putDonation()

```
putDonation (id, donation) {
  console.log('REQUESTING ' + donation._id + ' ' +
    JSON.stringify(donation, null, 5))
  return Api().put(`/donations/${id}`, donation,
    { headers: {'Content-type': 'application/json'} })
}
```

```
DonationService.putDonation(this.$router.params, donation)
  .then(response => {
    console.log(response)
    console.log('AFTER PUT ' + JSON.stringify(donation, null, 5))
  })
  .catch(error => {
    this.errors.push(error)
    console.log(error)
  })
```

postDonation()

```
postDonation (donation) {
  return Api().post('/donations', donation,
    { headers: {'Content-type': 'application/json'} })
},
```

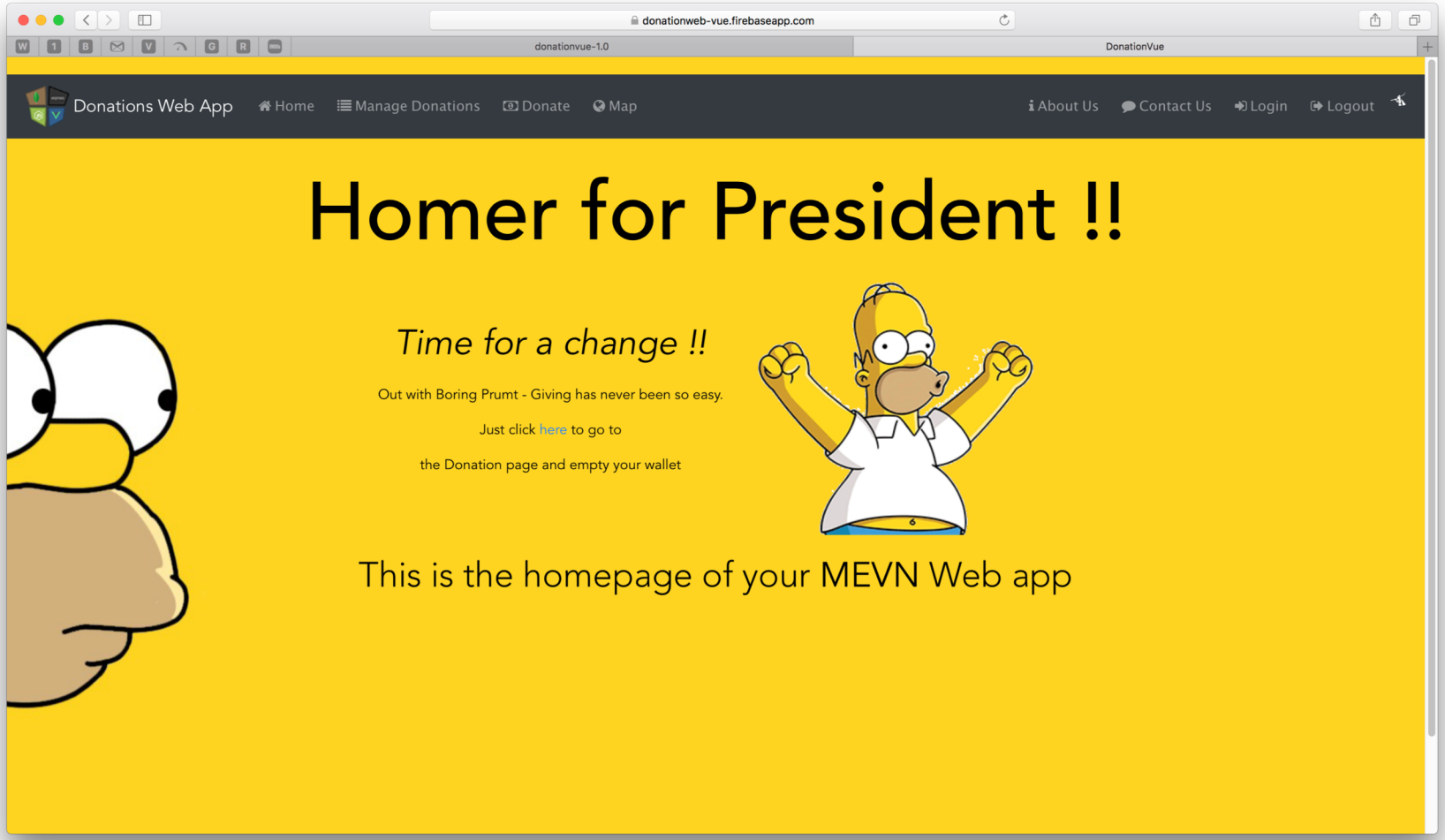
setting Headers

```
DonationService.postDonation(donation)
  .then(response => {
    console.log(response)
  })
  .catch(error => {
    this.errors.push(error)
    console.log(error)
  })
```

Case Study

LABS IN ACTION

Demo Application https://donationweb-vue.firebaseapp.com



References

- https://vuejs.org