



Frontend

Martijn Koch

////////////////////

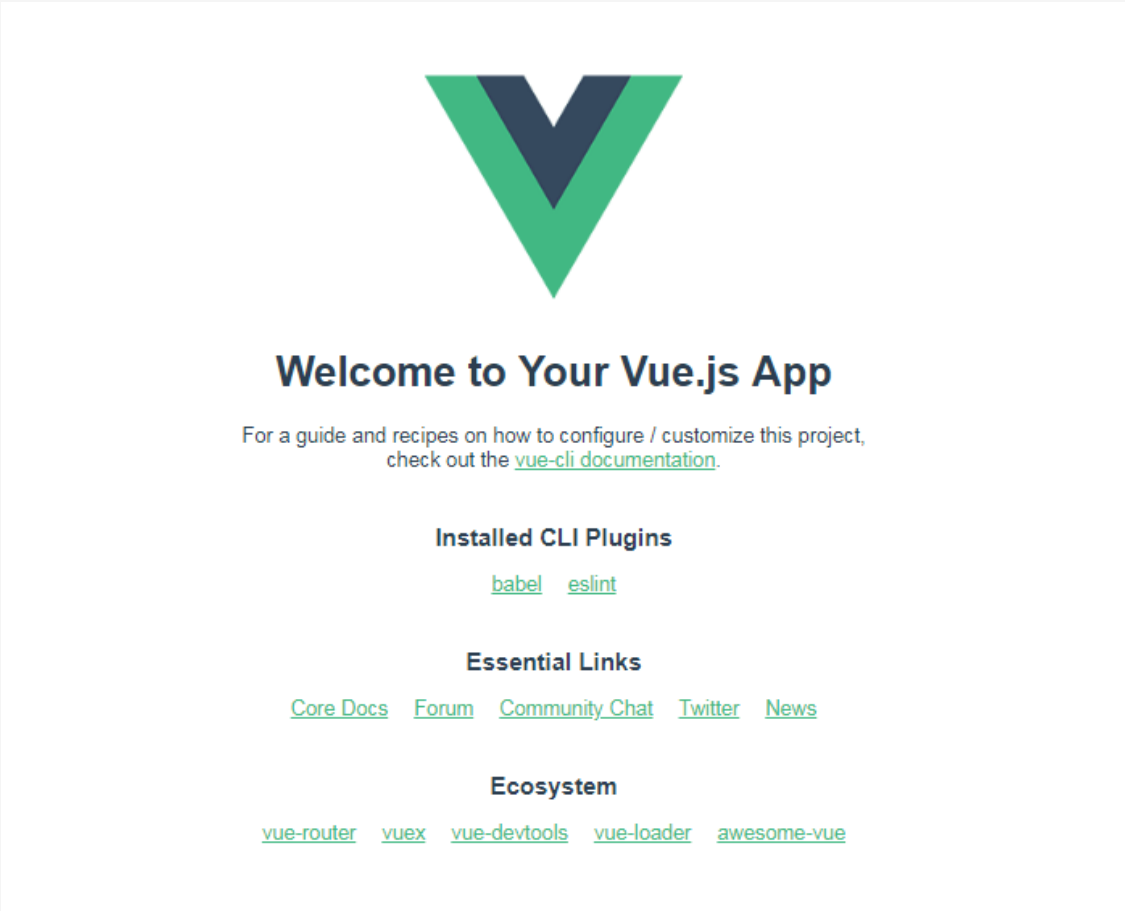
Frontend

Na overleg met het development team is de keuze aan de frontend komen te vallen op Vue.js. Vue.js is een populair Javascript framework en wordt veel in samenwerking met Laravel gebruikt. Zo zegt Grzybowska (2019) dat Vue.js op dit moment het meest populaire Javascript framework is en tevens is Javascript de meest populaire programmeertaal. Verder is Vue.js vrij gebruiksvriendelijk. Het heeft een goede compiler en er is ondersteuning van webpack. Vanwege de populariteit van Vue.js is er veel informatie over te vinden.

Net zoals bij de backend worden hieronder een aantal belangrijke zaken toegelicht en er zijn een aantal code snippets ter ondersteuning.

Gebruiker registreren

Om Vue.js te installeren moet je gebruik maken van Vue CLI. Dit installeer je met NPM, een Javascript package manager. Vue CLI is een command line tool die er voor zorgt dat er een aantal tools met Vue worden geïnstalleerd voor sneller development. Je hoeft dus niet alles van een nulpunt op te bouwen maar je kan sneller beginnen met bouwen van de applicatie. Als Vue CLI succesvol is geïnstalleerd krijg je de volgende startpagina te zien.



Frontend css library

Voor Vue.js zijn er een aantal populaire frontend css libraries. Elke library heeft een eigen design uitgangspunt. Een frontend library is gekozen om ook het development proces te versnellen. Gekozen is voor de library: BootstrapVue. Deze library biedt veel components aan, die out of the box kunnen worden gebruikt.

Als je een library aan een Vue.js applicatie wil toevoegen dan moeten deze geïnstalleerd worden door middel van NPM. Vervolgens moet je de library toevoegen aan je main.js file.

```
1  import Vue from 'vue'
2  import App from './App.vue'
3  import vueHeadful from 'vue-headful';
4  import BootstrapVue from 'bootstrap-vue'
5  import axios from 'axios'
6  import { library } from '@fortawesome/fontawesome-svg-core'
7  import { faUserSecret, faSearch } from '@fortawesome/free-solid-svg-icons'
8  import { FontAwesomeIcon } from '@fortawesome/vue-fontawesome'
9
10 library.add(faUserSecret, faSearch);
11 Vue.component('vue-headful', vueHeadful);
12 Vue.component('font-awesome-icon', FontAwesomeIcon)
13
14 Vue.config.productionTip = false;
15 Vue.prototype.$http = axios;
16
17 import router from './router/index.js'
18 import 'bootstrap/dist/css/bootstrap.css'
19 import 'bootstrap-vue/dist/bootstrap-vue.css'
```

Routing

Om routing te verzorgen in een Javascript framework zijn er verschillende routing libraries. De gebruikelijke router library voor vue is: Vue Router. Met Vue Router is het gemakkelijk om single page applicaties te maken. De Router is ook toegevoegd aan de main.js file en vervolgens kunnen in een router bestand de paden naar pagina's worden vastgesteld.

```
16  export default new Router({
17    mode: 'history',
18    routes: [
19      {
20        path: '/',
21        name: 'Login',
22        component: Login
23      },
24      {
25        path: '/dashboard',
26        name: 'Dashboard',
27        component: Dashboard
28      },
29      {
30        path: '/games',
31        name: 'GameOverview',
32        component: GameOverview
33      },
34      {
35        path: '/profile',
36        name: 'Profile',
37        component: Profile
38      },
39      {
40        path: '/games/:id',
41        name: 'GameDetail',
42        component: GameDetail
43      },
```

In de router geef je aan wat het pad zal zijn van de webpagina. Wat de naam gaat zijn en welke component er moet worden geladen. In Vue.js wordt net als in andere Javascript frameworks gebruik gemaakt van components. Door middel van components is het makkelijk om stukken code ergens anders te hergebruiken. De components van je applicatie laad je bovenin het router bestand in.

```
1  import Vue from 'vue'
2  import Router from 'vue-router'
3  import Dashboard from '@components/Dashboard'
4  import GameOverview from '@components/GameOverview'
5  import GameDetail from '@components/GameDetail'
6  import AdSetup from '@components/AdSetup'
7  import SetupGuide from '@components/SetupGuide'
8  import SetupGuides from '@components/SetupGuides'
9  import Profile from '@components/Profile'
10 import Touchportal from '@components/Touchportal'
11 import Obs from '@components/Obs'
12 import Login from '@components/Login'
```

HTTP client

Om de Laravel Api aan de achterkant aan te spreken en gebruik te maken van de data en data te posten, moet er gebruik worden gemaakt van een HTTP Client. Tot 2018 was Vue-Resource de ondersteunde HTTP Client voor Vue.js. Maar You (2018) meldde dat Vue-Resource niet meer de geadviseerde HTTP Client is. Deze is sindsdien Axios. Dit is een HTTP Client die in elke Javascript omgeving kan worden gebruikt. Net zoals andere libraries is deze geïnstalleerd met NPM.

Om data uit de API te halen moet met Axios een GET request worden uitgevoerd. In de Mounted functie van een Vue component kan een request worden vastgesteld. Op de pagina waar alle beschikbare games voor de streamer staan wordt de volgende GET request gedaan.

```
59  mounted: function() {
60    this.$http
61    .get('http://localhost:8888/api/games')
62    .then(response => {
63      console.log(response)
64      this.games = response.data})
65    .catch(error => this.games = [{title: 'no games found'}]);
66  },
```

Om de data daadwerkelijk op de pagina te vertonen in plaats van alleen in de console moet middels een for loop door de data worden gelopen. Daarna kan de data worden ingeladen in de Vue components.

```
<b-col lg="4" md="6" sm="12" class="mt-5 pb-5" v-for="game in games" v-bind:key="game.id">
  <b-card
    :title="game.name"
    :img-src="'http://localhost:8888/'+game.image"
    img-alt="Image"
    img-top
    style="max-width: 20rem;"
    class="mb-2 default-card hvr-grow col-lg-12 col-md-12 col-sm-12 mr-5"
  >
    <div class="row card-content">
      <div class="lg-6">
        <b-card-text v-if="game.ads_count > 1">{{game.ads_count}} ads available</b-card-text>
        <b-card-text v-else>{{game.ads_count}} ad available</b-card-text>
      </div>
      <div class="lg-6">
        <router-link v-bind:to="/game/'+game.id'">
          <b-button class="float-right homepage-cardbutton text-center" v-if="game.ads_count > 1">See ads</b-button>
          <b-button class="float-right homepage-cardbutton text-center" v-else>See ad</b-button>
        </router-link>
      </div>
    </div>
  </b-card>
</b-col>
```

Op de pagina van een game worden meerdere GET requests gedaan binnen een functie. Hiermee worden verschillende advertenties en scènes binnengehaald. Ook wordt met Axios het id van een specifieke game meegegeven aan de URL. In de Vue Router image is te zien dat het pad /games/:id is. Door middel van Axios wordt het juiste id meegegeven, waardoor de GET request kan worden uitgevoerd.

```
118 mounted: function() {
119   // Get the first scene
120   this.$http
121     .get(`http://localhost:8888/api/games/${this.$route.params.id}/scene-one`)
122     .then(response => {
123       console.log(response)
124       this.sceneOne = response.data})
125     .catch(error => this.sceneOne = [{title: 'no scene found'}]);
126
127   // Get the First ads
128   this.$http
129     .get(`http://localhost:8888/api/games/${this.$route.params.id}/ads-one`)
130     .then(response => {
131       console.log(response)
132       this.adsOne = response.data})
133     .catch(error => this.adsOne = [{title: 'no ads found'}]);
134
135   // Get the second scene
136   this.$http
137     .get(`http://localhost:8888/api/games/${this.$route.params.id}/scene-two`)
138     .then(response => {
139       console.log(response)
140       this.sceneTwo = response.data})
141     .catch(error => this.sceneTwo = [{title: 'no scene found'}]);
142
143   // Get the Second ads
144   this.$http
145     .get(`http://localhost:8888/api/games/${this.$route.params.id}/ads-two`)
146     .then(response => {
147       console.log(response)
148       this.adsTwo = response.data})
149     .catch(error => this.adsTwo = [{title: 'no ads found'}]);
150
151   // Get the third scene
152   this.$http
153     .get(`http://localhost:8888/api/games/${this.$route.params.id}/scene-three`)
154     .then(response => {
155       console.log(response)
156       this.sceneThree = response.data})
157     .catch(error => this.sceneThree = [{title: 'no scene found'}]);
158
159   // Get the third ads
160   this.$http
161     .get(`http://localhost:8888/api/games/${this.$route.params.id}/ads-three`)
162     .then(response => {
163       console.log(response)
164       this.adsThree = response.data})
165     .catch(error => this.adsThree = [{title: 'no ads found'}]);
166 },
167 }
```

Bronnen

Vue.js. Version 2.6.10[Framework] - The Progressive JavaScript Framework. (2019)
Geraadpleegd van
<https://vuejs.org/>

Grzybowska, k. (2019, 16 september). Top 10 most popular JavaScript frameworks in 2019. Geraadpleegd
<https://divante.com/blog/top-10-popular-javascript-frameworks-2019/>

Webpack. Version 4.41.5[Library] - A bundler for javascript and friends. (2019)
Geraadpleegd van
<https://webpack.js.org/>

Vue CLI. Version 4.1.2[Library] - Standard Tooling for Vue.js Development. (2019)
Geraadpleegd van
<https://cli.vuejs.org/>

NPM. Version 12.14.0[Software] - Build amazing things. (2019) Geraadpleegd van
<https://www.npmjs.com/>

BootstrapVue. Version 2.1.0[Library] (2019) Geraadpleegd van
<https://bootstrap-vue.js.org/>

Vue Router. Version 3.1.3[Library] (2019) Geraadpleegd van
<https://router.vuejs.org/>

You, E. (2018, 20 juni). Retiring vue-resource. Geraadpleegd van
<https://medium.com/the-vue-point/retiring-vue-resource-871a82880af4>

Axios. Version 0.19.0[Library] (2019) Geraadpleegd van
<https://github.com/axios/axios>

Vue.js Mounted. Version 2.6.10[Framework] (2019) Geraadpleegd van
<https://vuejs.org/v2/api/#mounted>