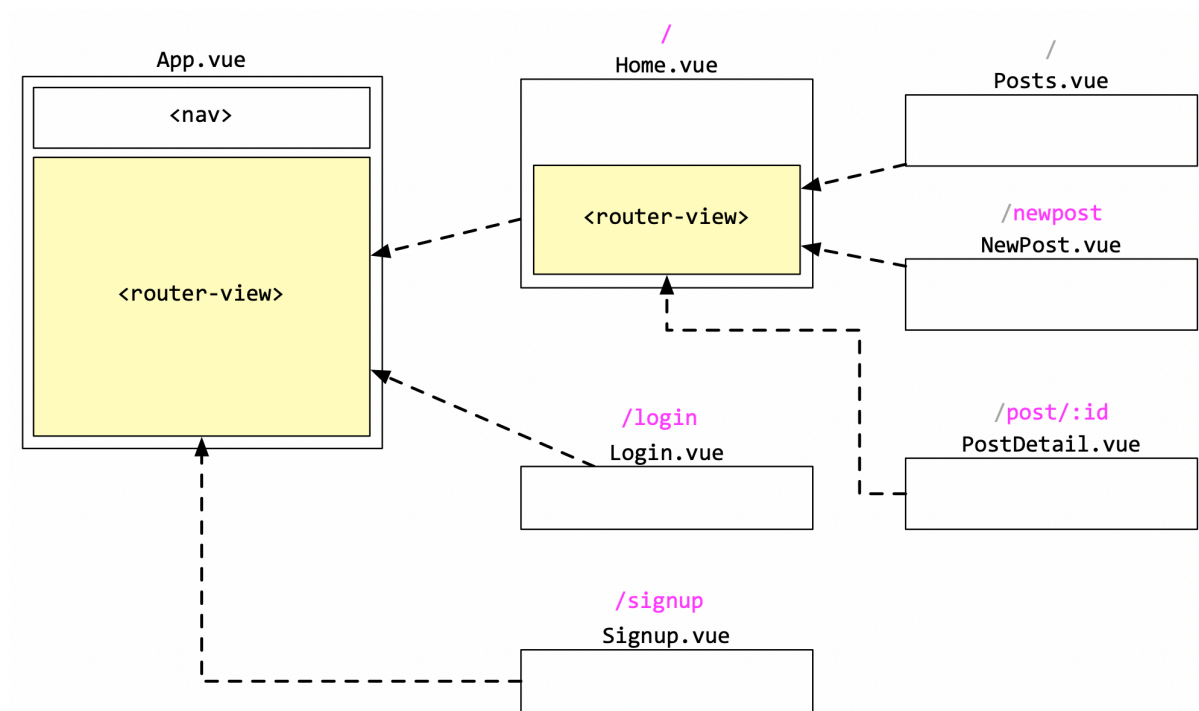


# Vue.js - Ugnježdene rute i podkolekcije

U ovoj vježbi prikazivat ćemo integraciju navigaciju kroz kompleksnije rute i prosljeđivanje ugnjeđenih podataka. Također prikazat ćemo na koji način možemo u Firebase-u spremiti vezane dokumente (veza 1 na više).

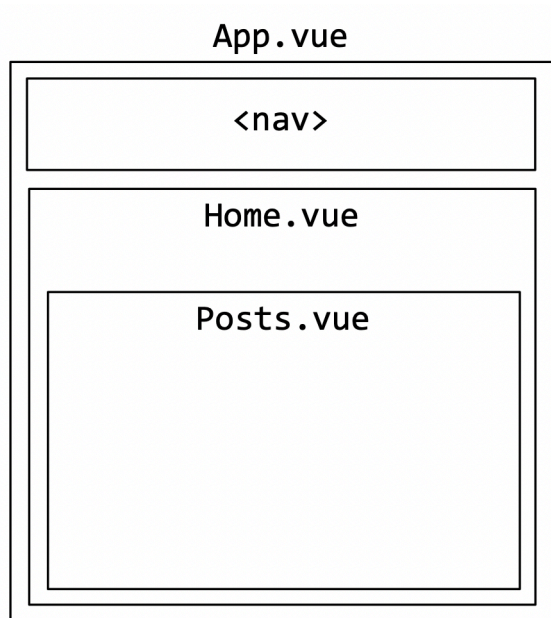
## Koraci

1. Kod sa prethodnih VUE-05 vježbi možemo preuzeti s GitHuba. Repozitorij: <https://github.com/fipu-na-stava/fipugram> (branch **step7**). Preuzimanje s Git-a, instaliranje paketa i pokretanje aplikacije pojašnjeno je u prethodnim vježbama.
2. U ovom ćemo koraku urediti korisničke rute kroz aplikaciju i zadužene komponente. To je jedan od važnijih koraka prilikom dizajna aplikacije. Preporuča se da svaka glavna funkcionalnost u aplikaciji ima vlastiti **endpoint, url ili rutu** (istoznačnice). To nam je važno kako bi naši korisnici mogli lako podijeliti url bilo kojeg dijela aplikacije.. Uredit ćemo rute Fipugram aplikacije na sljedeći način:



Slika 1. Korisničke rute kroz aplikaciju

Glavna i početna ruta kroz aplikaciju jest početna ruta: "/" ili zavisno o konfiguraciji i mjestu izvođenja: " <http://localhost:8080/>". Iz slike 1. je vidljivo kako će ta ruta prikazivati tri ugnježdene komponente:



Slika 2. Konačan izgled početne rute

Konfiguracijom rutera pomoću `router/index.js` datoteke projekta možemo postići navedeni kompoziciju (no prije toga potrebno je stvoriti nove prazna komponente `NewPost.vue` , `PostDetail.vue` , `Posts.vue` ):

```
1  //...
2  const routes = [
3    {
4      path: '/',
5      name: 'home',
6      component: Home,
7      children: [
8        {
9          path: '',
10         name: 'posts',
11         component: Posts
12       },
13       {
14         path: 'newpost',
15         name: 'newpost',
16         component: () => import(/* webpackChunkName: "post" */
17         '../views/NewPost.vue')
18       },
19       {
20         path: 'post/:id',
21         props: true, // važno da nam se puni "id" kao parametar komponente
22         "PostDetail.vue"
23         name: 'post-detail',
24         component: () => import(/* webpackChunkName: "post-detail" */
25         '../views/PostDetail.vue')
26       }
27     ]
28   },
29   {
30     path: '/login',
31     name: 'login',
```

```

29     component: () => import(/* webpackChunkName: "login" */
    './views/Login.vue')
30   },
31   {
32     path: '/signup',
33     name: 'signup',
34     component: () => import(/* webpackChunkName: "signup" */
    './views/Signup.vue')
35   },
36 ]
37 //...

```

Komponente u ruti mogu biti unaprijed učitane (npr. `Home` i `Posts` komponenta) ili dinamički učitane na zahtjev (npr. ostale komponente pomoću `import` metode). Rezultat je isti, razlika je u tome što nam je važnije i u kojem trenutku želimo učitati dodatne komponente (odmah pri startu ili kasnije kada ih korisnik zatraži).

3. Izdvojiti ćemo funkcionalnost listanja postova ( `InstagramCard` koji ima `v-for` ) u komponentu `Posts.vue` te u mjesto koda za unos novog posta postaviti `Button` koji vodi na stranicu `/newpost`.

```

1   <template>
2     <div v-if="authenticated">
3       <button @click="newImage()" type="Novi post" class="btn btn-primary
ml-2">
4         Post new image
5       </button>
6       <div @click="gotoDetails(card)" :key="card.id" v-for="card in
filteredCards">
7         <InstagramCard :info="card"/>
8       </div>
9     </div>
10  </template>
11
12  <script>
13    import InstagramCard from "@/components/InstagramCard.vue";
14    import store from "@/store.js";
15
16    export default {
17      data() {
18        return store;
19      },
20      computed: {
21        filteredCards() {
22          return this.cards.filter(card =>
card.title.includes(this.searchTerm));
23        }
24      },
25      name: "posts",
26      methods: {
27        gotoDetails(card) {
28          this.$router.push({path: `post/${card.id}`})

```

```

29     },
30     newImage() {
31       this.$router.push({name: 'newpost'}).catch(err => console.log(err))
32     }
33   },
34   components: {
35     InstagramCard
36   },
37 }
38 </script>
39
40 <style scoped>
41   button {
42     margin-bottom: 20px
43   }
44 </style>

```

Sadržaj komponente `Posts.vue`

Primjetimo nekoliko bitnih stvari. Ruta za prikaz detalja pojedinog posta ima i parametar `post/:id` te samim time u metodi `gotoDetails(card)` moramo navesti točan `id` posta u `this.$router.push({path: 'post/${card.id}'})`. Dakle vidimo kako možemo imati *dinamičke* rute koje primaju parametre. Zavisno o parametru `id` ruta `post/:id` prikazivat će točno određeni instaclone post.

Time je sadržaj `Home.vue` komponente značajno reduciran na samo `<router-view/>`:

```

1   <template>
2     <div class="row">
3       <div class="col-8">
4         <router-view/>
5       </div>
6       <div class="col-4"></div>
7     </div>
8   </template>
9
10  <script>
11    import store from "@store.js";
12
13    export default {
14      data() {
15        return store;
16      },
17      name: "home"
18    };
19  </script>

```

Sadržaj komponente `Home.vue`

- Sljedeći korak je implementacija komponente za unos novog posta na ruti `/newpost` u koju ćemo prebaciti sve relevantne dijelove iz bivšeg `Home.vue` vezane uz upload nove slike:

```

1   <template>

```

```

2     <form @submit.prevent="postImage" class="mb-5">
3         <croppa :width="400" :height="400" v-model="imageData"></croppa>
4         <button type="submit" class="btn btn-primary ml-2">Post image</button>
5     </form>
6 </template>
7
8 <script>
9 import store from "@/store.js";
10
11 export default {
12     data() {
13         return store
14     },
15     name: "newpost",
16     methods: {
17         postImage() {
18             this.imageData.generateBlob(blobData => {
19                 if (blobData != null) {
20                     // ako koristimo "/" u nazivu slike, Storage fino napravi
21                     // Konkretno u ovom primjeru imat ćemo direktorij nazvan po
22                     // Slika će biti nazvana po trenutnom vremenu kako bi imali
23                     // jedinstveni naziv slike.
24                     let imageName = this.userEmail + "/" + Date.now() + ".png";
25
26                     storage
27                         .ref(imageName)
28                         .put(blobData)
29                         .then(result => {
30                             result.ref.getDownloadURL()
31                             .then(url => {
32                                 db.collection("posts")
33                                 .add({
34                                     email: this.userEmail,
35                                     posted_at: Date.now(),
36                                     url: url
37                                 })
38                                 .then(docRef => {
39                                     console.log("Document written with ID: ",
40                                     docRef.id);
41
42                                     this.imageData = null;
43                                     this.$router.push({name: "posts"})
44                                 })
45                                 .catch(e => {
46                                     console.error("Error adding document: ", error);
47                                 });
48                             })
49                             .catch(e=> {
50                                 console.error(e)
51                             })
52                         })
53                 }
54             });
55         }
56     }
57 }

```

```

50         .catch(e => {
51             console.error(e)
52         })
53     }
54     }); // da... zatvaranje zagrada nakon ovoga noćna je mora!
55 }
56 }
57 };
58 </script>
59
60 <style lang="scss">
61
62     .card-body {
63         padding: 0px;
64     }
65
66     img:hover {
67         cursor: pointer;
68     }
69
70     .btn-post {
71         width: 200px;
72     }
73
74     .croppa-container {
75         border: 3px dashed gray;
76         width: 400px;
77         display: block;
78         margin-bottom: 10px;
79     }
80 </style>

```

5. U ovom koraku implemenitrati ćemo komponentu `PostDetail.vue` koje će zasebno prikazati samo jedan post zajedno s njegovim komentarima. Prvo modificirat ćemo komponentu `InstagramCard.vue` na način da prikazuje komentare i dopušta unos novog komentara. Komentare ćemo spremiti u podkolekciju prethodne kolekcije `posts` :

```

1     <template>
2
3         <div class="card text-center">
4             <div class="card-header text-left">
5                 By <strong>{{ info.email }}</strong>
6             </div>
7             <div class="card-body">
8                 
9             </div>
10            <div class="card-footer text-left">
11                <div>{{ timeAgo }}</div>
12
13            <!-- ⬇ ⬇ NOVO ⬇ ⬇ -->

```

```

14         <div v-if="showcomments">
15             <div class="comments list-group">
16                 <a :key="c.posted_at" v-for="c in comments" href="#"
class="animate list-group-item list-group-item-action flex-column align-
items-start">
17                     <div class="d-flex w-100 justify-content-between">
18                         <small>{{ formatTime(c.posted_at) }} by {{ c.email }}
</small>
19                     </div>
20                     <small>{{ c.comment }}</small>
21                 </a>
22             </div>
23
24             <form @submit.prevent="postComment" class="form-inline mb-5">
25                 <div class="form-group">
26                     <input v-model="newComment" type="text" class="form-
control" id="imageUrl" placeholder="Any comment?">
27                 </div>
28                 <button type="submit" class="btn btn-primary ml-
2">Post</button>
29             </form>
30         </div>
31         <!-- ⬆ ⬆ NOVO ⬆ ⬆ -->
32
33     </div>
34 </div>
35
36 </template>
37
38 <script>
39     import moment from 'moment'
40     import store from '@/store.js'
41
42     export default {
43         props: [ "info", "showcomments" ],
44         data () {
45             return {
46                 global: store,
47                 newComment: "",
48                 comments: []
49             }
50         },
51         // ⬇ ⬇ ⬇ ⬇ NOVO ⬇ ⬇ ⬇ ⬇
52         mounted() {
53             if (this.showcomments) {
54                 // pristup podkolekciji, koja mora za svog roditelja imati konkretni
dokument nadkolekcije
55                 // u ovom slučaju: komentari su dio konkretnog instaclone posta
56
57                 db.collection("posts").doc(this.info.id).collection("comments").onSnapshot
(snapshot => {
                    snapshot.docChanges().forEach(change => {

```

```

58         if (change.type === "added") {
59             let comment = change.doc.data();
60             comment.isNew = true;
61             console.log(comment);
62             this.comments.unshift(comment)
63         }
64     })
65 })
66 }
67 },
68 methods: {
69     formatTime(t) {
70         return moment(t.posted_at).fromNow()
71     },
72     postComment() {
73         if (this.newComment) {
74             db.collection("posts") // kolekcija roditelj
75             .doc(this.info.id) // konkretni post
76             .collection("comments") // podkolekcija
77             .add({ email: this.global.userEmail, comment:
this.newComment, posted_at: Date.now() })
78             .then(result => {
79                 console.log(result)
80                 this.newComment = "";
81             })
82             .catch(e => {
83                 console.error(e)
84             })
85         }
86     }
87 },
88 // 🚀 🚀 🚀 🚀 NOVO 🚀 🚀 🚀 🚀
89 computed: {
90     timeAgo () {
91         return moment(this.info.posted_at).fromNow()
92     }
93 }
94 }
95 </script>
96
97 <style>
98     .card {
99         margin: 10px
100     }
101     .comments {
102         margin: 20px 0
103     }
104
105     /* -----
106     * Generated by Animista on 2020-1-3 19:19:35
107     * Licensed under FreeBSD License.
108     * See http://animista.net/license for more info.

```



```

109      * w: http://animista.net, t: @cssanimista
110      * ----- */
111      @-webkit-keyframes fade-in-fwd{0%{-webkit-
transform:translateZ(-80px);transform:translateZ(-80px);opacity:0}100%{-
webkit-
transform:translateZ(0);transform:translateZ(0);opacity:1}}@keyframes
fade-in-fwd{0%{-webkit-
transform:translateZ(-80px);transform:translateZ(-80px);opacity:0}100%{-
webkit-transform:translateZ(0);transform:translateZ(0);opacity:1}}
112      .animate{-webkit-animation:fade-in-fwd 1s cubic-
bezier(.39,.575,.565,1.000) both;animation:fade-in-fwd 1s cubic-
bezier(.39,.575,.565,1.000) both}
113
114      </style>

```

Zatim ćemo oblikovati komponentu `PostDetail.vue` koja koristi proširenu komponentu `InstagramCard` :

```

1      <template>
2          <div v-if="card">
3              <InstagramCard :info="card" :showcomments="true"/>
4          </div>
5      </template>
6
7      <script>
8          import InstagramCard from "@components/InstagramCard.vue";
9          import store from "@store.js";
10
11      export default {
12          props: ['id'],
13          data() {
14              return {
15                  card: null
16              }
17          },
18          mounted() {
19              console.log(this.id);
20              db.collection("posts")
21                  .doc(this.id).get().then(doc => {
22                      this.card = doc.data();
23                      this.card.id = doc.id;
24                  })
25          },
26          name: "post-detail",
27          components: {
28              InstagramCard
29          },
30      }
31      </script>
32
33      <style scoped>
34          button {

```

```
35     margin-bottom: 20px
36   }
37 </style>
```

Važno je da komponenti `InstagramCard` proslijedimo parametar `:showcomments="true"` kako bi prikazivala komentare. Druga važna stvar je da ova komponenta ima `props: ["id"]` gdje se parametar `id` puni iz putanje rute npr. `http://localhost:8080/post/QL5JBGHrmpwh2WgiAf8P` dakle `id = QL5JBGHrmpwh2WgiAf8P`. Vue automatski puni taj parametar za nas jer smo u ruti postavili `props = true` (vidi Korak 2.)