

Vue EventManager

A Vue EventManager App

Vue EventManager

Let us now train our knowledge a bit with an event manager app.

Target

Which adventures do you expect?

- Sightseeing
- Shopping
- Zoo visits
- Fine dining
- Jewellery & Watches Museum

ZÜRICH



Zürich is the largest city in Switzerland and the capital of the canton of Zürich. It is located in north-central Switzerland, at the northwestern tip of Lake Zürich. As of January 2020, the municipality has 434,335 inhabitants, the urban area (agglomeration) 1.315 million (2009), and the Zürich metropolitan area 1.83 million (2011). Zürich is a hub for railways, roads, and air traffic. Both Zurich Airport and Zürich's main railway station are the largest and busiest in the country.

Adventures:

- Sightseeing
- Shopping
- Zoo visits
- Fine dining

GENEVA



Geneva is the second-most populous city in Switzerland (after Zürich) and the most populous city of Romandy, the French-speaking part of Switzerland. Situated where the Rhône exits Lake Geneva, it is the capital of the Republic and Canton of Geneva.

Adventures:

- Sightseeing
- Shopping
- Fine dining
- Jewellery & Watches Museum

BASEL



Basel is a city in northwestern Switzerland on the river Rhine. Basel is Switzerland's third-most-populous city (after Zürich and Geneva) with about 175,000 inhabitants. The official language of Basel is (the Swiss variety of Standard) German, but the main spoken language is the local Basel German dialect.

Adventures:

- Sightseeing
- Zoo visits

Vue EventManager

Let us create two interfaces:

```
1 import { ref } from 'vue';
2
3 interface adventure {
4   id: number,
5   adventure: string,
6   options: string[]
7 }
8
9 interface canton {
10  id: number,
11  canton: string,
12  image: string,
13  description: string,
14  adventures: number[]
15 }
```

Vue EventManager

Start the setup() function with basic data:

```
1  setup() {  
2    const EVENTMANAGER_BASE = 'http://localhost:3000';  
3    const adventureList = ref<adventure[]>([]);  
4    const selectedAdventure = ref<number[]>([]);  
5    const filteredCantonList = ref<canton[]>([]);  
6    let cantonList: canton[] = [];  
7  
8  
9    // next code here  
10 }
```

Vue EventManager

Create a generateFilteredCantonList

```
1  function generateFilteredCantonList(): void {
2    filteredCantonList.value = [];
3
4    for (const canton of cantonList) {
5      const cantonAdventuresArray = canton.adventures;
6      const selectedAdventuresArray = selectedAdventure.value;
7
8      if (hasAllAdventures(cantonAdventuresArray, selectedAdventuresArray)) {
9        filteredCantonList.value.push(canton);
10     }
11   }
12 }
```

Vue EventManager

Create a hasAllAdventures function:



```
1  function hasAllAdventures(cantonAdventures: number[], selectedAdventures: number[]): boolean {  
2    return selectedAdventures.every(f => cantonAdventures.includes(f));  
3  }
```

Vue EventManager

Create a getAdventureList function:

```
1 // Get the list of skills from the Express API
2 async function getAdventureList(): Promise<void> {
3     const response = await fetch(`${EVENTMANAGER_BASE}/adventures`);
4     adventureList.value = await response.json();
5 }
```


Vue EventManager

Create a getCantonList function:

```
1 // Get the list of applications from the Express API
2 async function getCantonList(): Promise<void> {
3     const response = await fetch(`${EVENTMANAGER_BASE}/cantons`);
4     cantonList = await response.json();
5     filteredCantonList.value = cantonList;
6 }
```

Vue EventManager

Call and return them:

```
1  getAdventureList();
2  getCantonList();
3
4  return {
5    adventureList,
6    selectedAdventure,
7    filteredCantonList,
8    generateFilteredCantonList,
9  }
```

Vue EventManager

Now we add up our html:



```
1 <div class="columns is-multiline mt-3">
2   <div v-for="canton in filteredCantonList" :key="canton.id" class="column is-one-third">
3     <div class="card">
```

Vue EventManager

Now we add up our html:

```
1 <header class="card-header">
2   <p class="card-header-title is-uppercase is-size-5">
3     {{ canton.canton }}
4   </p>
5 </header>
```

Vue EventManager

Now we add up our html:



```
1 <figure class="image is-256x256">
2   
3 </figure>
```

Vue EventManager

Now we add up our html:

```
1 <p class="is-size-9 mb-4">{{ canton.description }}</p>
2     <h4>Adventures:</h4>
3     <ul v-for="adventure in canton.adventures" :key="adventure.id">
4         <li>
5             <p>{{ adventureList[adventure-1].adventure }}</p>
6         </li>
7     </ul>
```

Possible Enhancements

Enhancement ideas (same component):

- Add a read more button / link to show the full text and only 100 characters first
- Add a button which colorizes the text
- Add a button which changes the shape of the picture
 - <https://bulma.io/documentation/elements/image/>
- ...

Possible Enhancements

Enhancement ideas (new component):

- Split the existing component in multiples
 - extract the text, image or the list of adventures
- Extract the Composition API functionality
- ...

Possible Enhancements

Now we have seen the Composition API in action.

It is still up to you to use it or not - but it can make complex components easier.

End



End

That was all for this chapter
