

# A Quest For A Storyteller's App

Adventures in the Lands of Electron and VueJs



# Katrin Kampfrath

@\_die\_katrin



@cssclasses

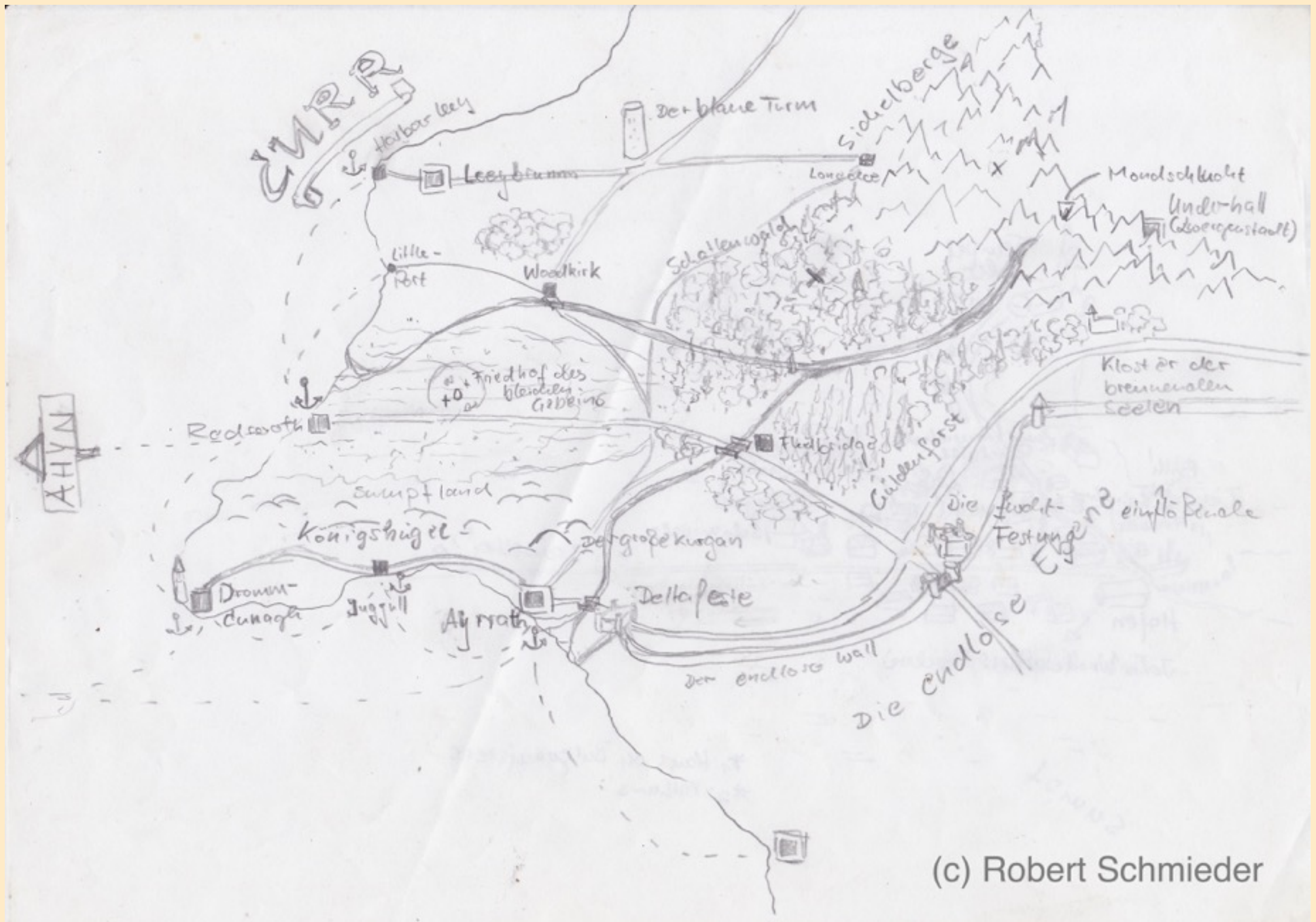
# A Quest For A Storyteller's App

Adventures in the Lands of Electron and VueJs

# The Prologue > Dungeons & Dragons



# The Prologue > Our Continent "Curr"



(c) Robert Schmieder





ELECTRON

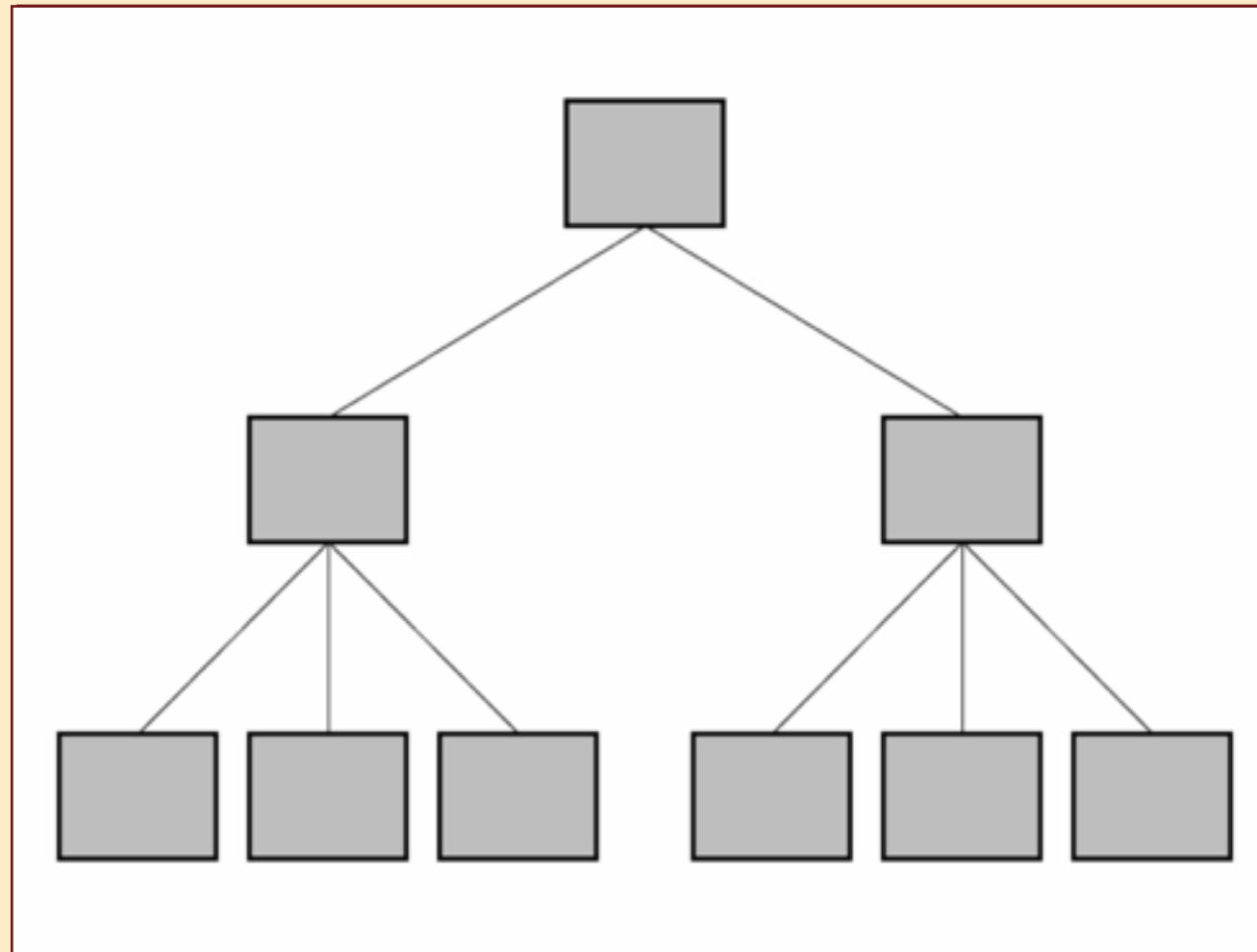


electron-vue

<https://github.com/SimulatedGREG/electron-vue>

# **“The 1000 Arms of The ‘Data Model’ Kraken”**

# Adventure > The Data Model



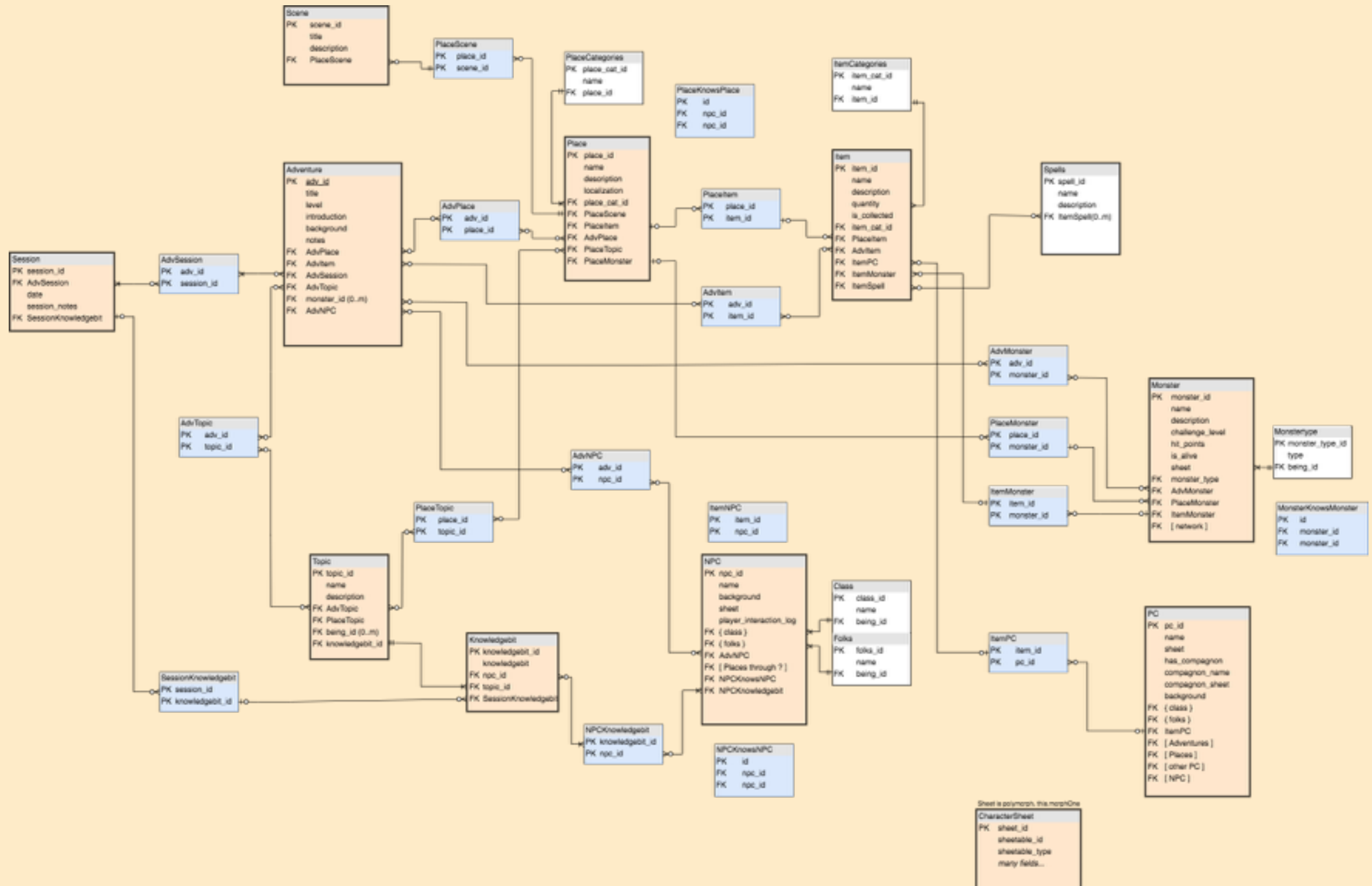
Hierarchical Model

Network Model

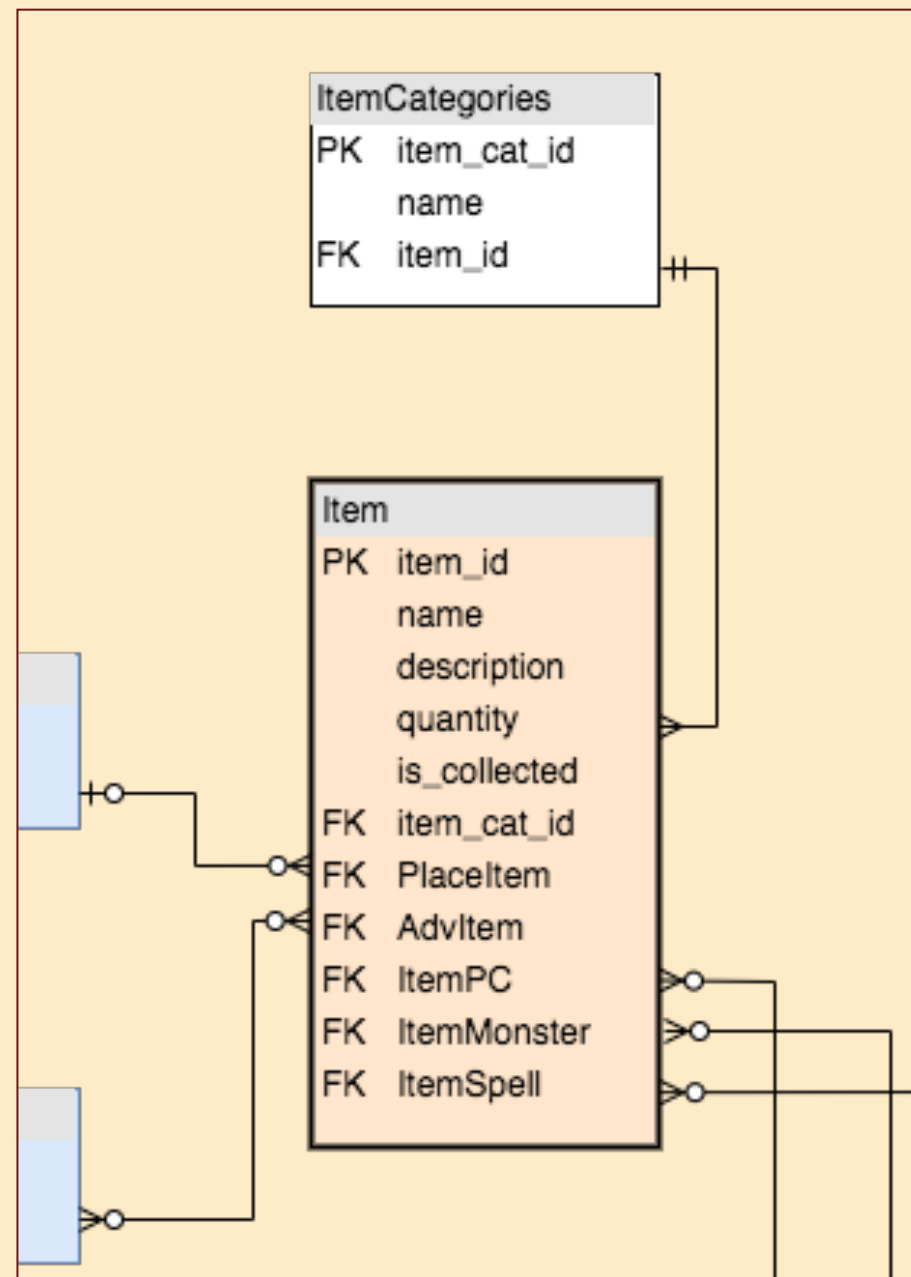
Entity-Relationship Model



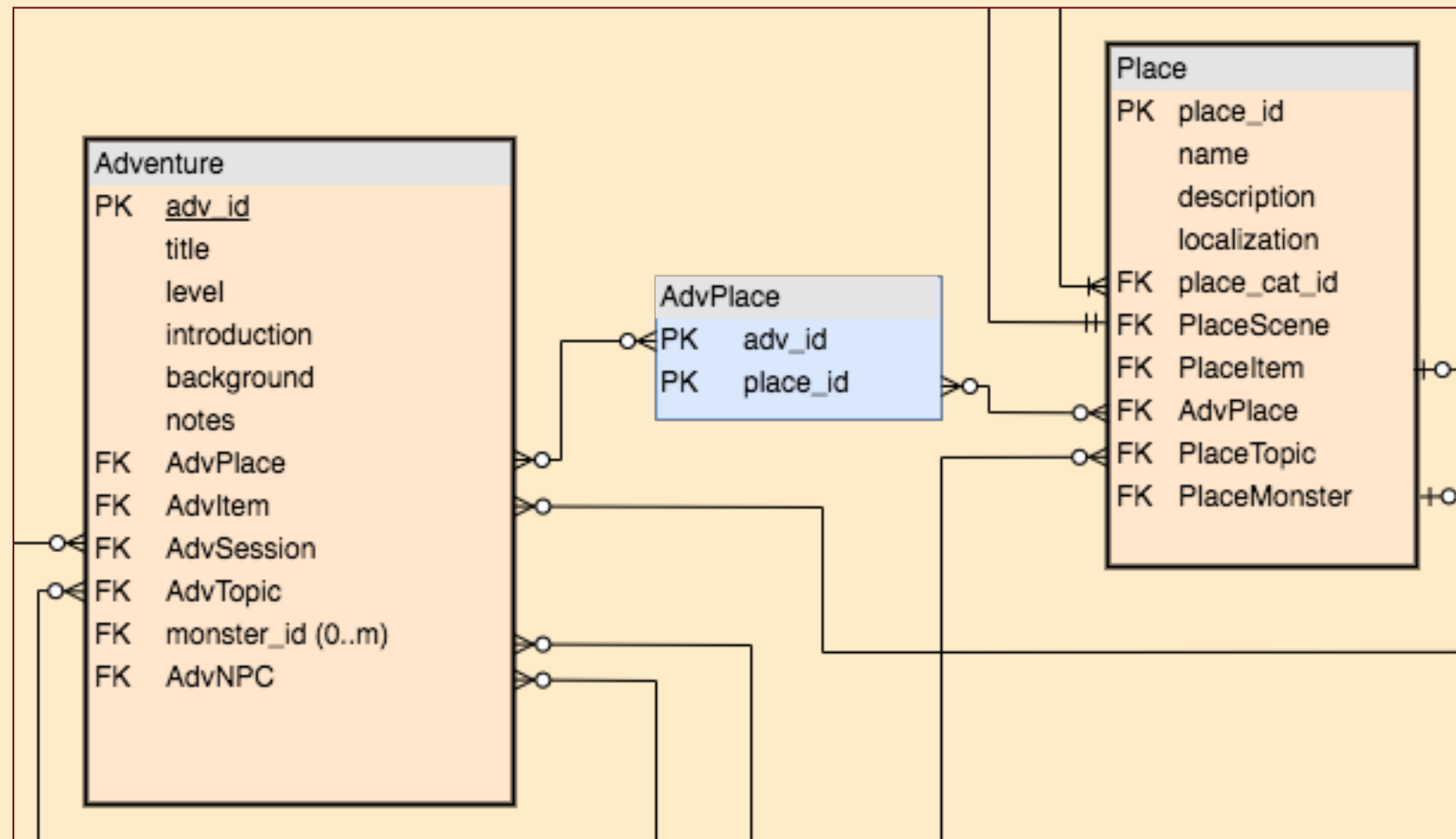
# Adventure > The Data Model



# Adventure > The Data Model



# Adventure > The Data Model



**“Forest of  
The Data Persistence”**

A good choice for any Electron app...



+



...yet, i diverted into a different direction



## **Vuex ORM**

<https://vuex-orm.github.io/vuex-orm/>



### **Vuex ORM**

- Plugin for Vuex
- adds Object-Relational Mapping onto Vuex
- has an `.increment()` Method



## Persistence

`createPersistedState` plugin from “vuex-electron”

## Adventure > The Data Persistence

### Vuex: store/index.js

```
import Vue from 'vue';
import Vuex from 'vuex';
import VuexORM from '@vuex-orm/core';
import { createPersistedState } from 'vuex-electron';

import database from '../models/index';
import modules from './modules';

Vue.use(Vuex);

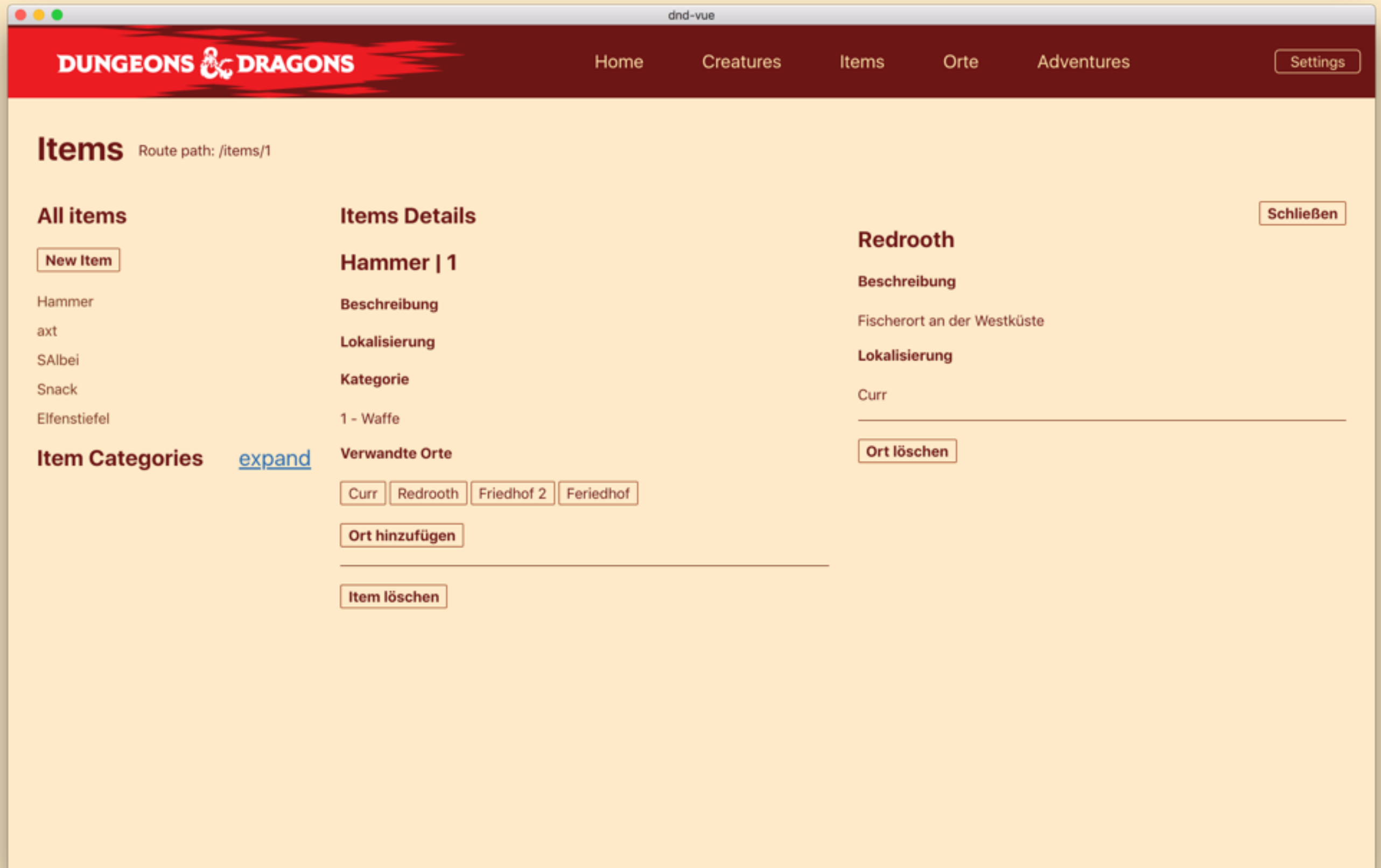
export default new Vuex.Store({
  modules,
  plugins: [VuexORM.install(database), createPersistedState()],
  strict: process.env.NODE_ENV !== 'production',
});
```

**“The Mage and The Sage  
Create a Multi-View Screen”**

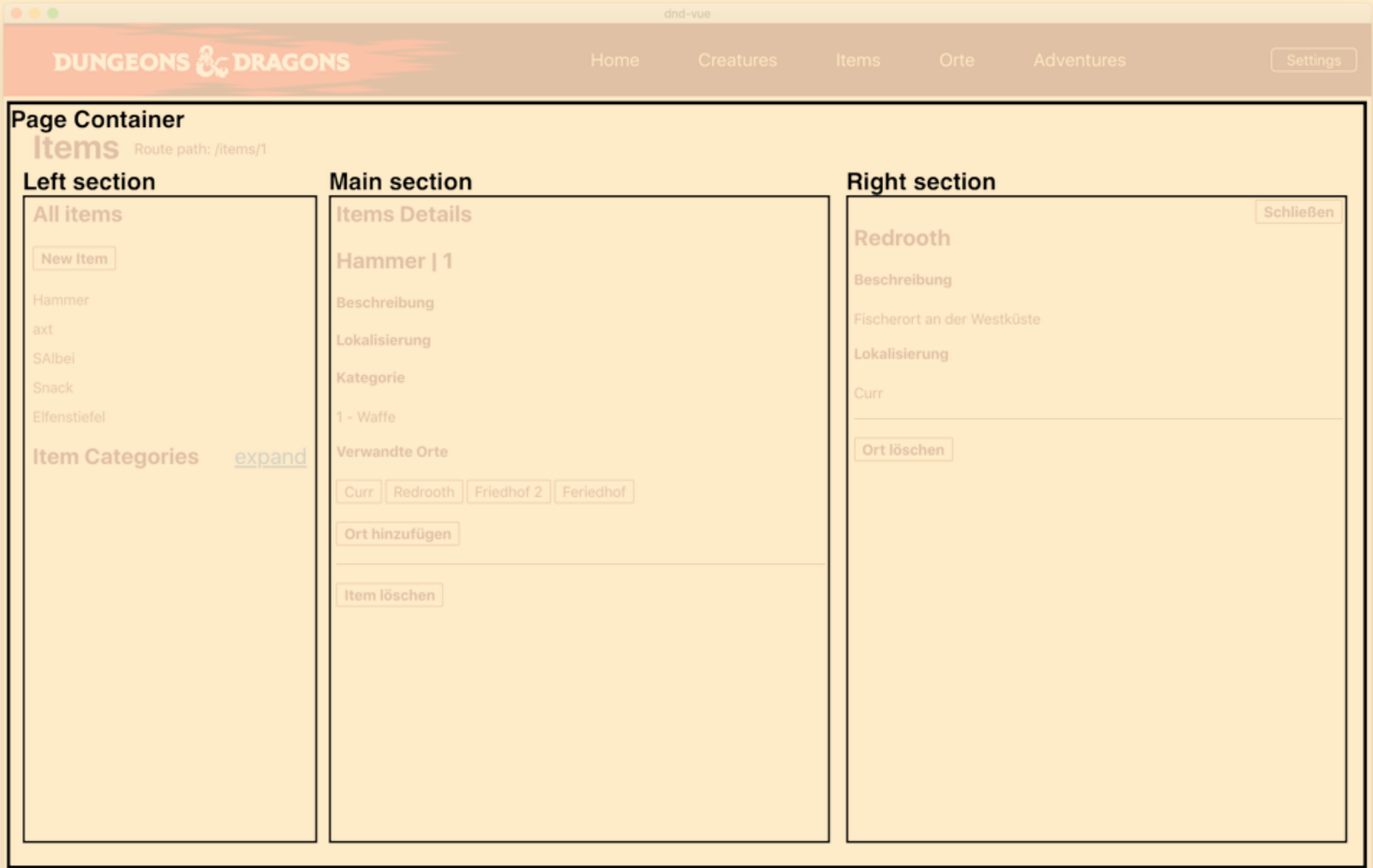
## **Pain points of our Dungeon Master**

- Missing quick access to information
- Looooots of information

# Adventure > The Data Display



# Adventure > The Data Display



## Page container: Item.vue

```
<template>
  <div class="bg-orange-200 wrapper wrapper--3col">
    <div class="wrapper--3col__top-row">
      <h1 class="text-4xl">Items</h1>
      <p class="mx-4">Route path: {{ $route.path }}</p>
    </div>
    <router-view name="sidebar" />
    <router-view name="main" />
    <DynamicSlot />
  </div>
</template>

<script>
import DynamicSlot from '../components/_shared/DynamicSlot';

export default {
  name: 'ItemsPage',
  components: {DynamicSlot}
};
</script>
```



## Page container: Item.vue

```
<template>
  <div class="bg-orange-200 wrapper wrapper--3col">
    <div class="wrapper--3col__top-row">
      <h1 class="text-4xl">Items</h1>
      <p class="mx-4">Route path: {{ $route.path }}</p>
    </div>
    <router-view name="sidebar" />
    <router-view name="main" />
    <DynamicSlot />
  </div>
</template>

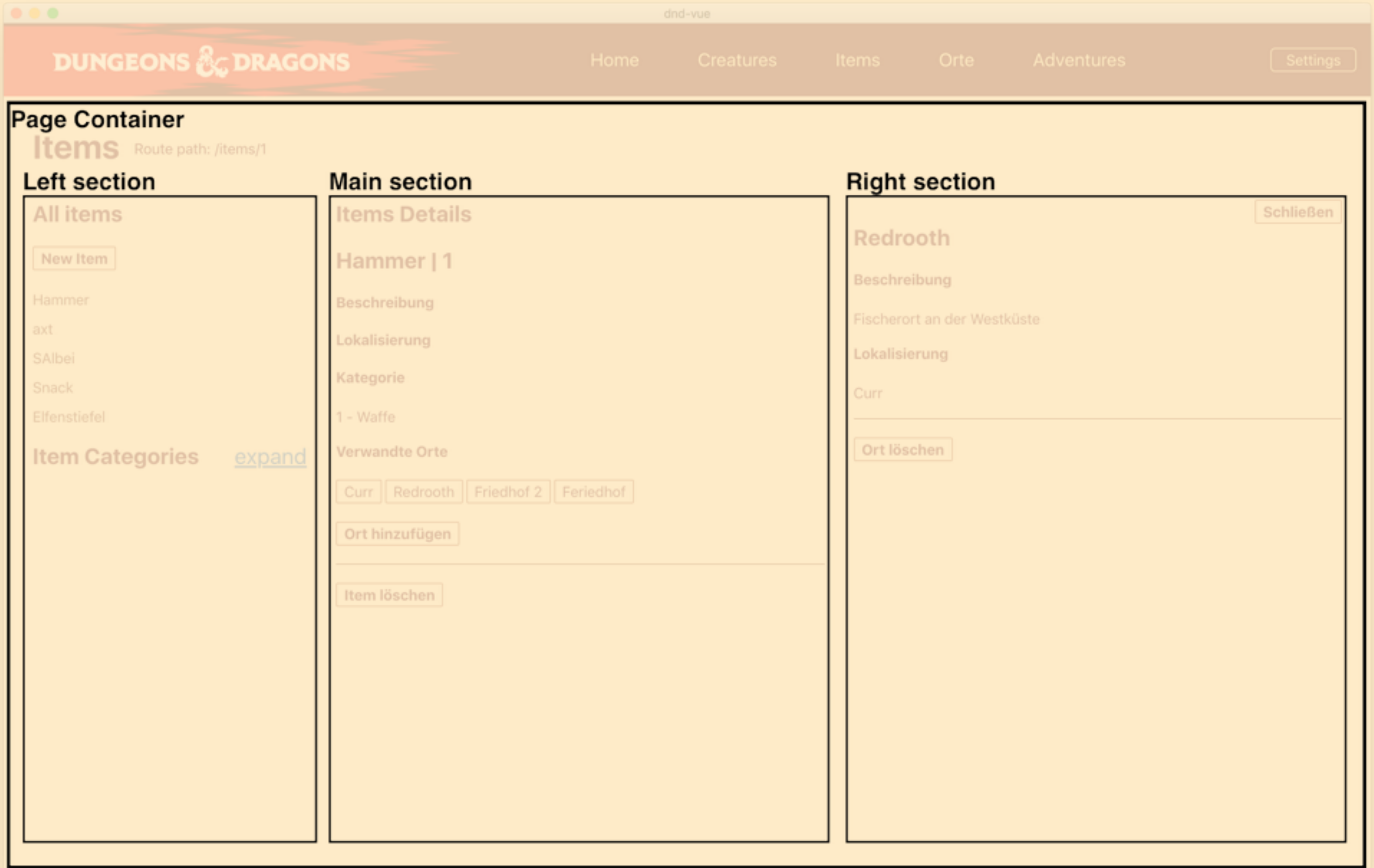
<script>
import DynamicSlot from '../components/_shared/DynamicSlot';

export default {
  name: 'ItemsPage',
  components: {DynamicSlot}
};
</script>
```

## Vue Router setup: router/index.js

```
{
  path: '/items',
  name: 'items-page',
  component: ItemsPage,
  children: [{
    path: ':id',
    name: 'items-details',
    components: {
      sidebar: ItemsIndex,
      main: ItemsDetail
    }
  }]
},
{
  path: 'places',
  name: 'places-page',
  component: PlacesPage,
  children: [{
    path: ':id',
    name: 'places-detail',
    components: {
      sidebar: PlacesIndex,
      main: PlacesDetail,
    }
  }]
},
```

# Adventure > The Data Display



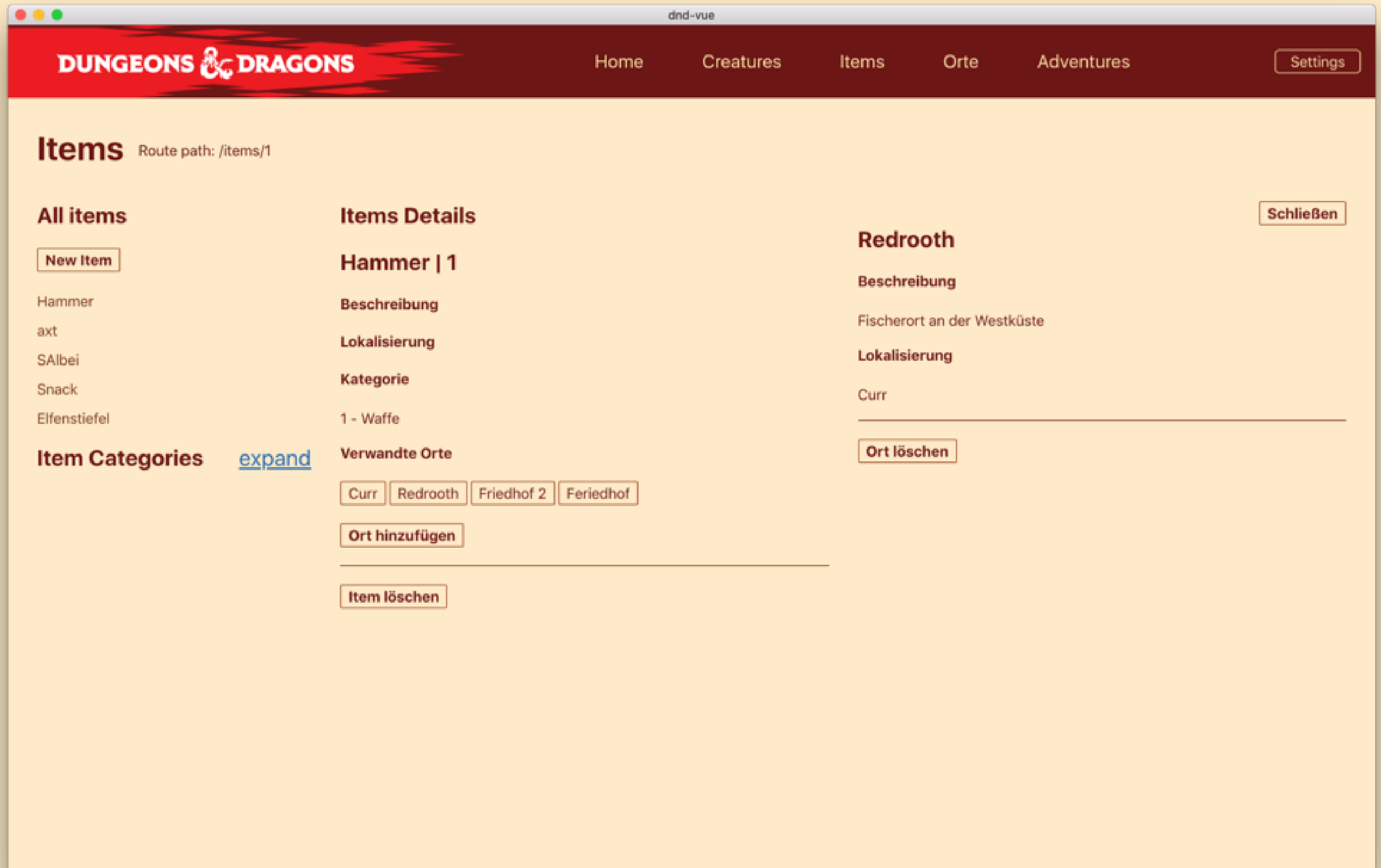
## Dynamic section: DynamicSlot.vue

```
<template>
  <aside>
    <Button v-if="currentComponent"
      text="Schließen"
      class="block ml-auto"
      @click.native="unloadComponent"
    />
    <component :is="currentComponent"
      v-if="currentComponent"
      :slot-id="slotId"
    />
  </aside>
</template>
```

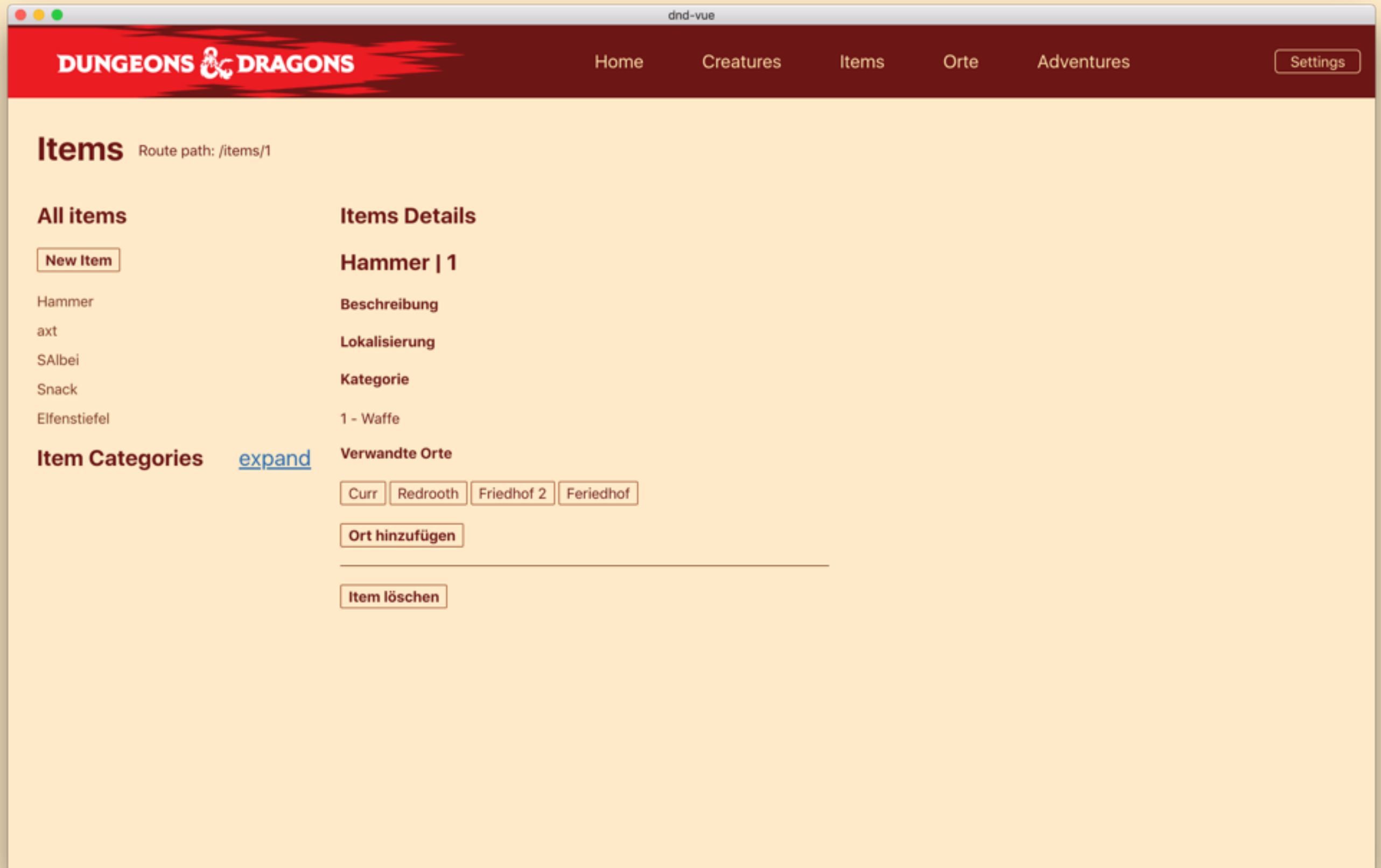
## Dynamic section: DynamicSlot.vue

```
<template>
  <aside>
    <Button v-if="currentComponent"
      text="Schließen"
      class="block ml-auto"
      @click.native="unloadComponent"
    />
    <component :is="currentComponent"
      v-if="currentComponent"
      :slot-id="slotId"
    />
  </aside>
</template>
```

# Adventure > The Data Display



# Adventure > The Data Display





## More adventures

- “Zombies vs. Vampires: Updating a 2-Year-Old Boilerplate”
- “Down In The Dungeon with ESLint and Prettier”
- “With a Little Help From My Golden Dragon: Find the Right Code Structure”
- Coming soon: “The Dancing Lights: Add a C++ Character Sheet”



**Thank you**

<https://github.com/katrin-k/dnd-stories>

[@\\_die\\_katrin](#)

<https://die-katrin.eu/frontend-dev>