

Міністерство освіти і науки України
Івано-Франківський національний технічний університет нафти і газу

Кафедра КСМ

Лабораторна робота №5
Тема “Робота з об’єктами та масивами”

Виконав студент
групи КІ-18-1
Чуйко Т.В.

Перевірив
Слабінога М.О.

м.Івано-Франківськ
2019р.

Мета: ознайомитися з функціями роботи з об'єктами та масивами, а також взаємодією об'єктів та їх відображення.

Умова задачі:

1. Згідно завдання, вибраного в роботі 3, реалізуйте таблицю з діями щодо записів, використовуючи об'єкти.
2. Окрім раніше реалізованих функцій додавання та видалення, реалізуйте функцію редагування.
3. Реалізуйте додавання, редагування та видалення зв'язаних даних.
4. Доповніть раніше зроблені AJAX-запити новими, згідно попередніх пунктів.

Весь код застосунку можна подивитися за посиланням:

<https://github.com/taraschuiko/web-labs/tree/master/Лаб5/bookmarks>

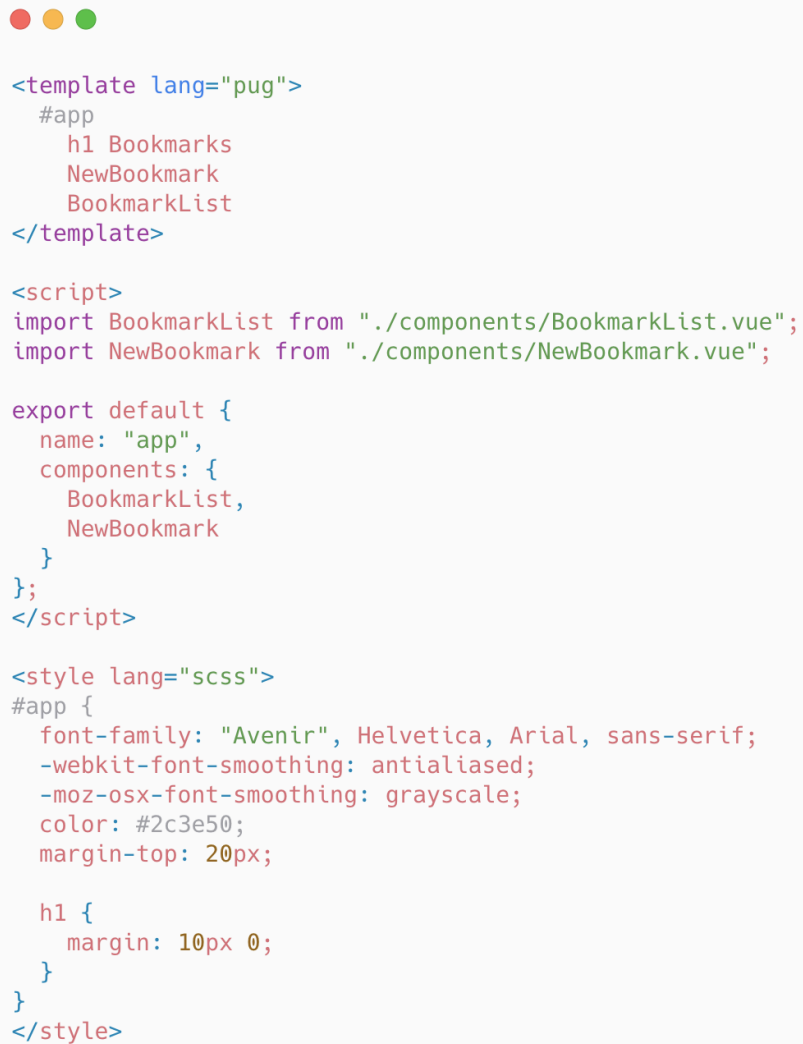
Сховище веб-застосунку:

```
import Vue from "vue";
import Vuex from "vuex";

Vue.use(Vuex);

export default new Vuex.Store({
  state: {
    bookmarks: [
      {
        id: 1431123123,
        name: "dfsdf",
        description: "ewrwerwe",
        link: "google"
      }
    ]
  },
  getters: {
    getBookmarks(state) {
      return state.bookmarks;
    }
  },
  mutations: {
    addBookmark(state, newBookmark) {
      state.bookmarks.push(newBookmark);
    },
    removeBookmark(state, id) {
      state.bookmarks = state.bookmarks.filter(bookmark => bookmark.id !== id);
    },
    editBookmark(state, data) {
      const index = state.bookmarks.findIndex(
        bookmark => bookmark.id === data.id
      );
      state.bookmarks.splice(index, 1, {
        name: data.name,
        description: data.description,
        link: data.link
      });
    }
  },
  actions: {
    addBookmark(context, data) {
      const newBookmark = {
        id: Math.floor(Math.random() * Math.floor(999999)),
        name: data.name,
        description: data.description,
        link: data.link
      };
      context.commit("addBookmark", newBookmark);
      fetch('https://137.0.0.1:3307/bookmarks', {
        method: "POST",
        body: JSON.stringify(newBookmark)
      });
    },
    removeBookmark(context, id) {
      context.commit("removeBookmark", id);
      fetch('https://137.0.0.1:3307/bookmarks/${id}', {
        method: "DELETE"
      });
    },
    editBookmark(context, data) {
      context.commit("editBookmark", data);
      fetch('https://137.0.0.1:3307/bookmarks/${data.id}', {
        method: "PATCH",
        body: JSON.stringify(data)
      });
    }
  }
});
```

Компонент App:



```
<template lang="pug">
  #app
    h1 Bookmarks
    NewBookmark
    BookmarkList
</template>

<script>
import BookmarkList from "../components/BookmarkList.vue";
import NewBookmark from "../components/NewBookmark.vue";

export default {
  name: "app",
  components: {
    BookmarkList,
    NewBookmark
  }
};
</script>

<style lang="scss">
#app {
  font-family: "Avenir", Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  color: #2c3e50;
  margin-top: 20px;

  h1 {
    margin: 10px 0;
  }
}
</style>
```

Компонент NewBookmark:

```

    ● ● ●

<template lang="pug">
  form.NewBookmark(@submit="addBookmark")
    label Name
      input(type="text" required v-model="name")
    label Description
      input(type="text" required v-model="description")
    label Link
      input(type="text" required v-model="link")
      input(type="submit" value="Add")
</template>

<script>
export default {
  name: "NewBookmark",
  data() {
    return {
      name: "",
      description: "",
      link: ""
    };
  },
  methods: {
    addBookmark(event) {
      event.preventDefault();
      this.$store.dispatch("addBookmark", {
        name: this.name,
        description: this.description,
        link: this.link
      });

      this.name = "";
      this.description = "";
      this.link = "";
    }
  }
};
</script>

<style lang="scss" scoped>
.NewBookmark {
  margin-bottom: 10px;

  label {
    display: inline-flex;
    flex-wrap: nowrap;
    margin: 5px 15px 5px 0;

    input {
      margin-left: 5px;
    }
  }
}
</style>
```

Компонент BookmarkList:



```
<template lang="pug">
  .BookmarkList
    a.BookmarkList__item(
      v-for="bookmark in bookmarks"
      :href="bookmark.link"
      target="_blank"
    )
      h2 {{ bookmark.name }}
      p {{ bookmark.description }}
</template>

<script>
export default {
  name: "BookmarkList",
  computed: {
    bookmarks() {
      return this.$store.getters.getBookmarks;
    }
  }
};
</script>

<style lang="scss" scoped>
.BookmarkList {
  &__item {
    display: block;
    margin-bottom: 5px;
    border: 1px solid #ccc;
    border-radius: 4px;
    padding: 5px 10px;

    h2,
    p {
      margin: 5px 0;
    }
  }
}
</style>
```

Компонент BookmarkList:

```
<template lang="pug">
  .Bookmark
    a.Bookmark__link(
      :href="bookmark.link"
      target="_blank"
    )
    h2 {{ bookmark.name }}
    p {{ bookmark.description }}
    button(@click="toggleEditing") Edit
    button(@click="removeBookmark") Remove
  .Bookmark__editing(v-if="isEditing")
    form.NewBookmark(@submit="editBookmark")
      label Name
      input(type="text" required v-model="newName")
      label Description
      input(type="text" required v-model="newDescription")
      label Link
      input(type="text" required v-model="newLink")
      input(type="submit" value="Update")
</template>

<script>
export default {
  name: "Bookmark",
  data() {
    return {
      isEditing: false,
      newName: this.bookmark.name,
      newDescription: this.bookmark.description,
      newLink: this.bookmark.link
    };
  },
  props: {
    bookmark: Object
  },
  methods: {
    toggleEditing(event) {
      event.preventDefault();
      this.isEditing = !this.isEditing;
      this.newName = this.bookmark.name;
      this.newDescription = this.bookmark.description;
      this.newLink = this.bookmark.link;
    },
    editBookmark(event) {
      event.preventDefault();
      this.$store.dispatch("editBookmark", {
        id: this.bookmark.id,
        name: this.newName,
        description: this.newDescription,
        link: this.newLink
      });
      this.toggleEditing();
    },
    removeBookmark(event) {
      event.preventDefault();
      this.$store.dispatch("removeBookmark", this.bookmark.id);
    }
  }
};
</script>

<style lang="scss" scoped>
.Bookmark {
  &__link,
  &__editing {
    display: block;
    margin-bottom: 5px;
    border: 1px solid #ccc;
    border-radius: 4px;
    padding: 5px 10px;
  }

  h2,
  p {
    margin: 5px 0;
  }

  &__editing {
    label {
      display: inline-flex;
      flex-wrap: nowrap;
      margin: 5px 15px 5px 0;
    }

    input {
      margin-left: 5px;
    }
  }
}
</style>
```

Висновок: На цій лабораторній роботі я ознайомився з функціями роботи з об'єктами та масивами, а також взаємодією об'єктів та їх відображення.