T5: Vue.js

Javiera Inostroza, Elías Sabja, Samuel Zúñiga

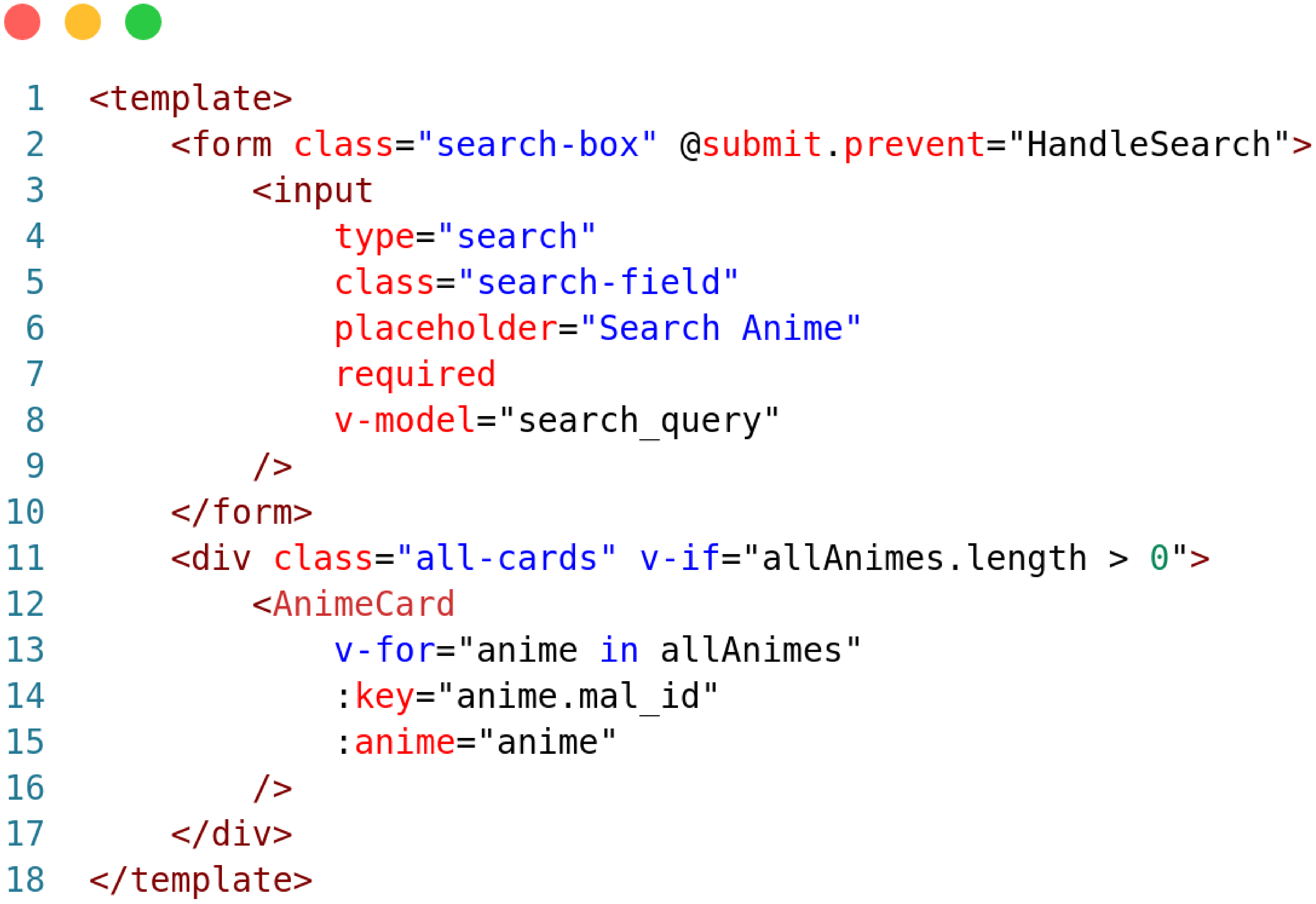
¿Qué hicimos?

MyAnimeList

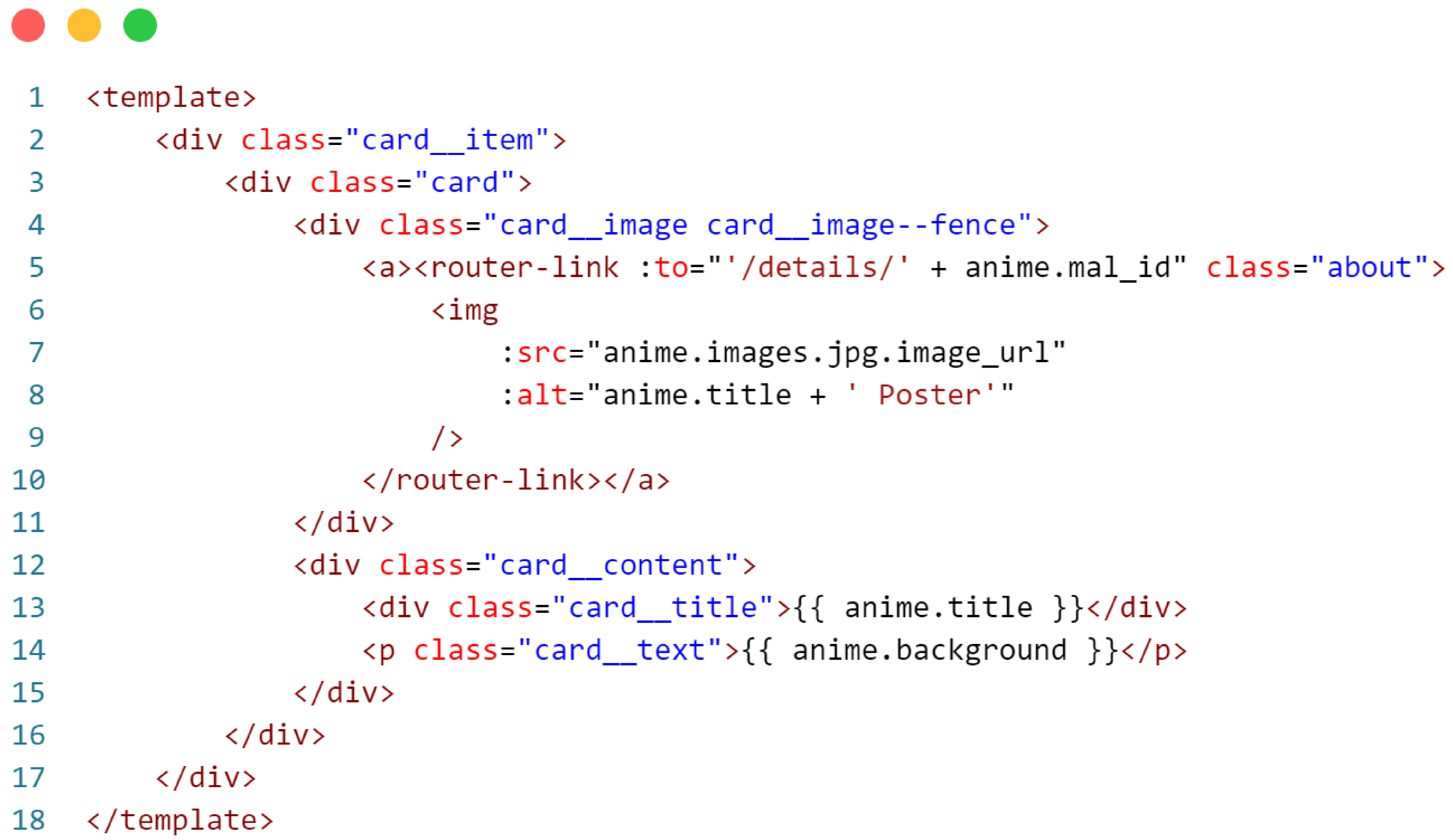
Jikan API

DEMO

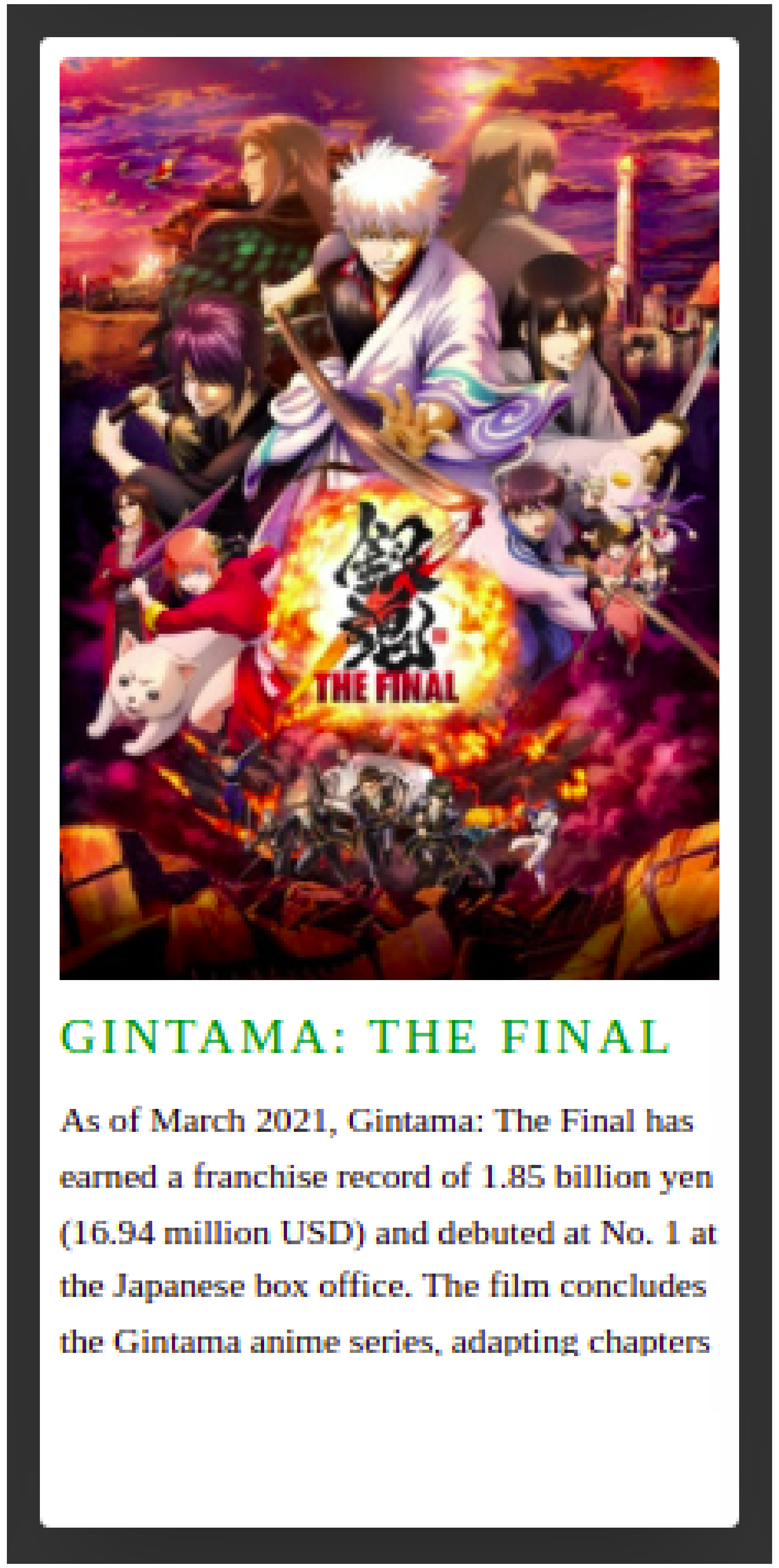
VUE



```
<template>
    <form class="search-box" @submit.prevent="HandleSearch">
        <input
            type="search"
            class="search-field"
            placeholder="Search Anime"
            required
            v-model="search_query"
        />
    </form>
    <div class="all-cards" v-if="allAnimes.length > 0">
        <AnimeCard
            v-for="anime in allAnimes"
            :key="anime.mal_id"
            :anime="anime"
        />
    </div>
</template>
```

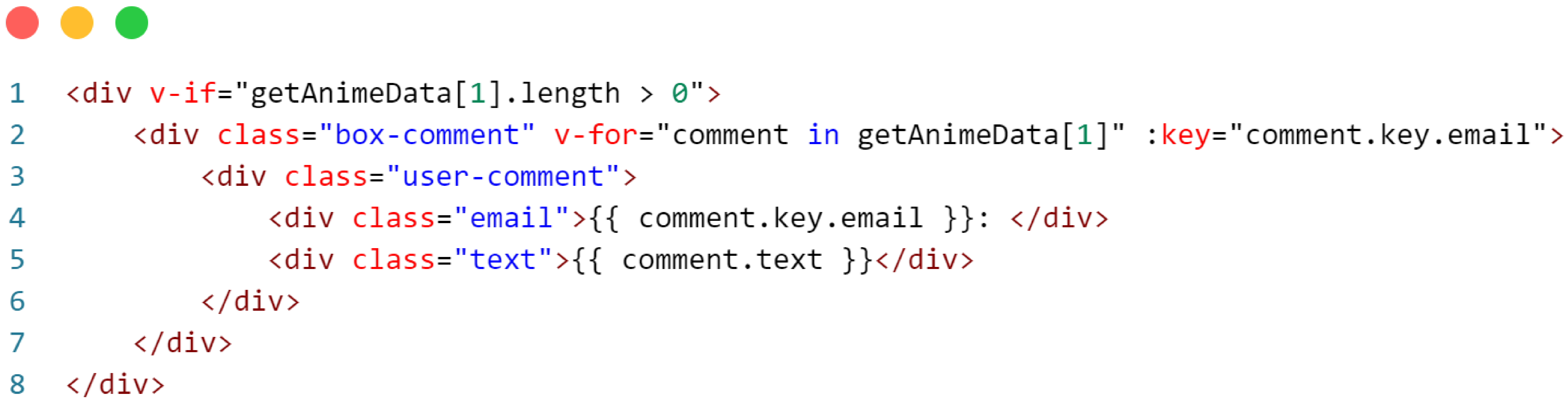


```
<template>
    <div class="card__item">
        <div class="card">
            <div class="card__image card__image--fence">
                <a><router-link :to="'/details/' + anime.mal_id" class="about">
                    <img
                        :src="anime.images.jpg.image_url"
                        :alt="anime.title + ' Poster'"
                    />
                </router-link></a>
            </div>
            <div class="card__content">
                <div class="card__title">{{ anime.title }}</div>
                <p class="card__text">{{ anime.background }}</p>
            </div>
        </div>
    </div>
</template>
```



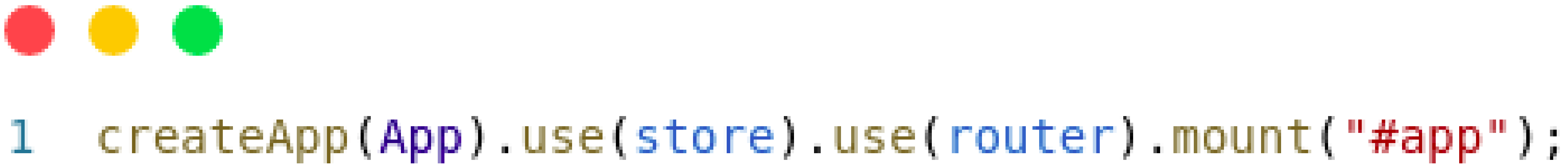
GINTAMA: THE FINAL

As of March 2021, Gintama: The Final has earned a franchise record of 1.85 billion yen (16.94 million USD) and debuted at No. 1 at the Japanese box office. The film concludes the Gintama anime series, adapting chapters

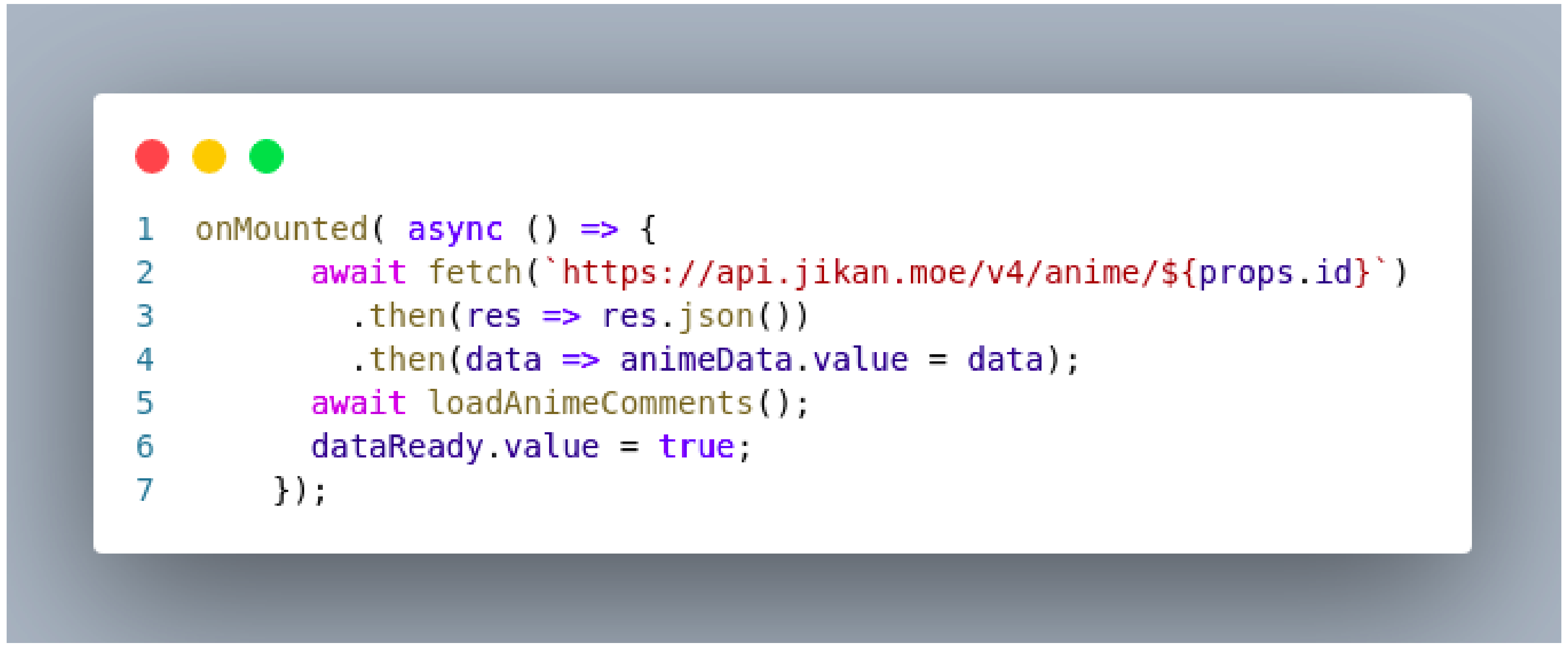


```
<div v-if="getAnimeData[1].length > 0">
    <div class="box-comment" v-for="comment in getAnimeData[1]" :key="comment.key.email">
        <div class="user-comment">
            <div class="email">{{ comment.key.email }}: </div>
            <div class="text">{{ comment.text }}</div>
        </div>
    </div>
</div>
```

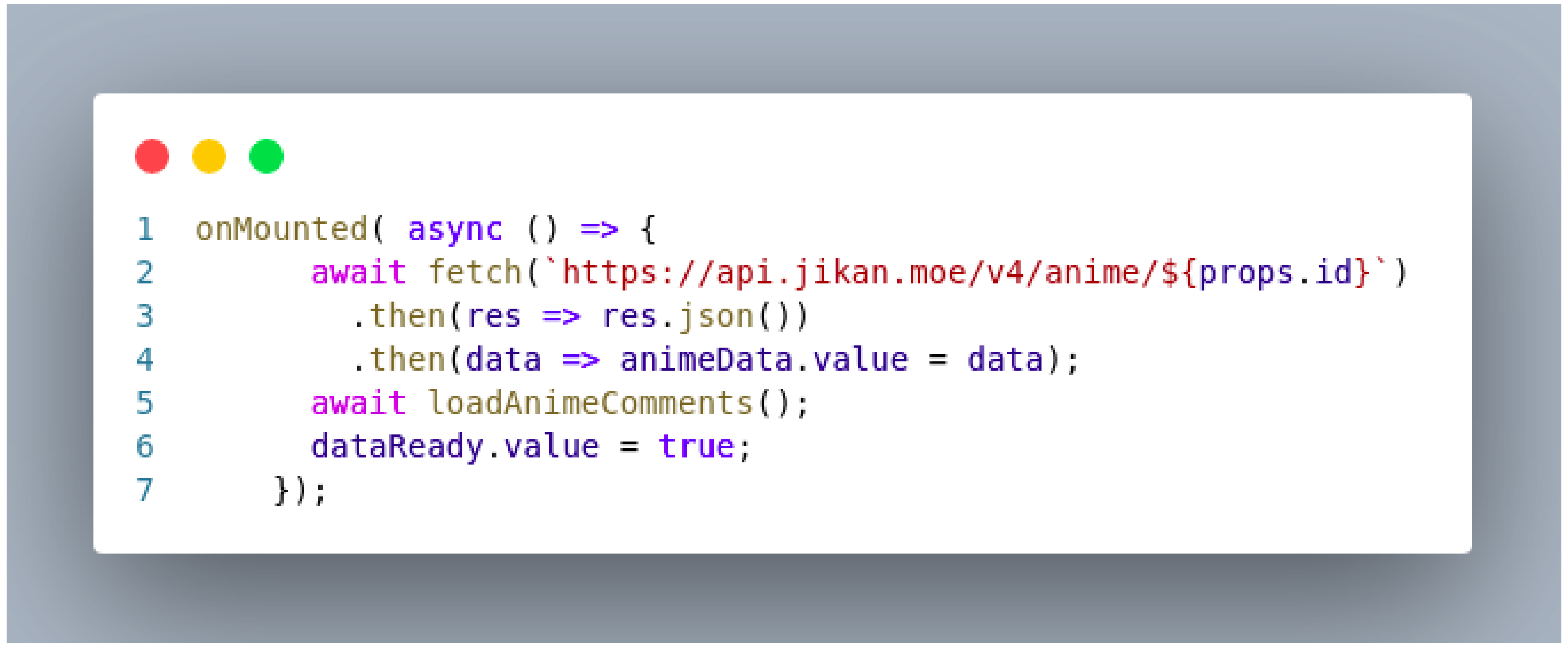
Main.js



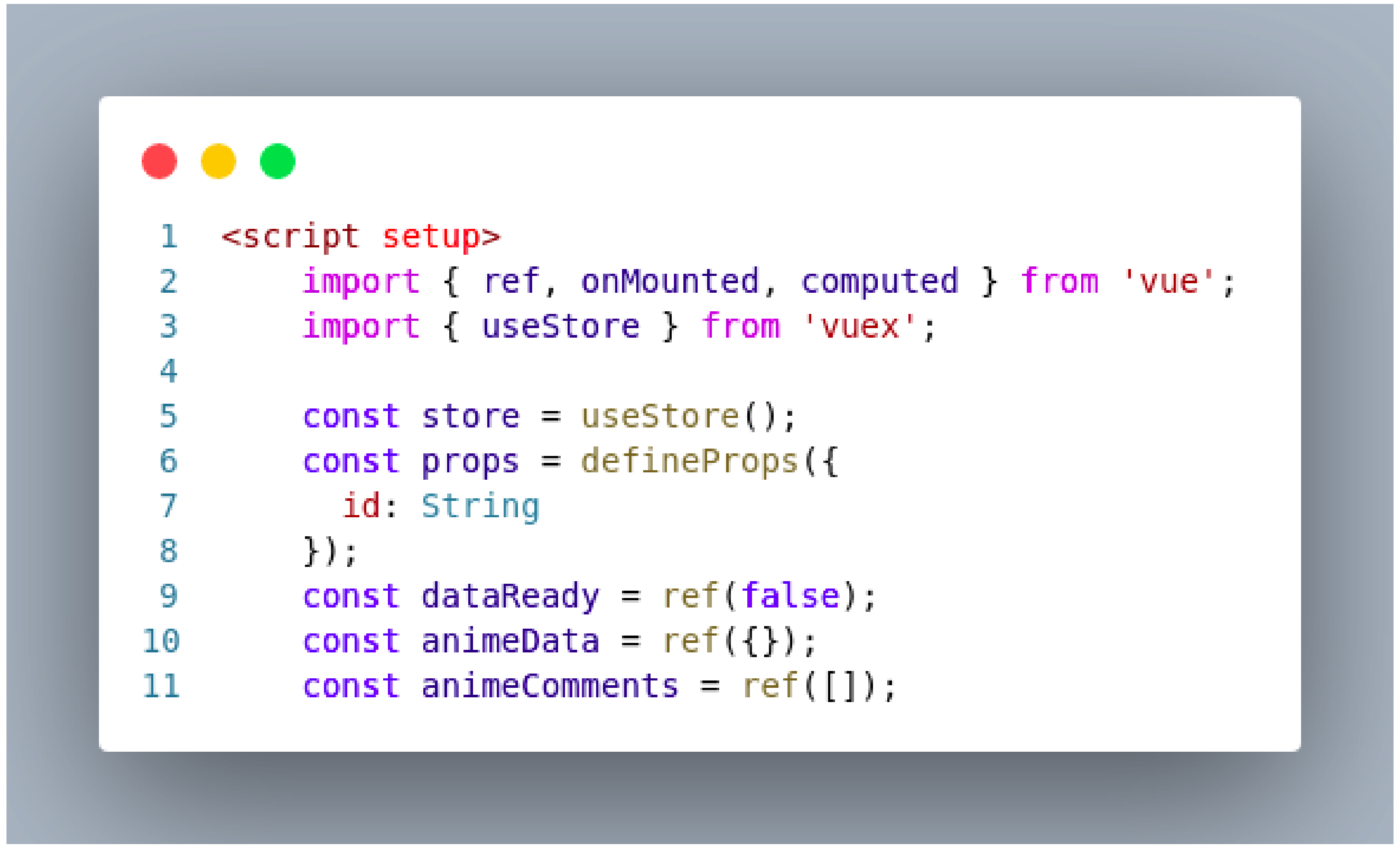
```
createApp(App).use(store).use(router).mount("#app");
```



```
onMounted( async () => {
      await fetch(`https://api.jikan.moe/v4/anime/${props.id}`)
        .then(res => res.json())
        .then(data => animeData.value = data);
      await loadAnimeComments();
      dataReady.value = true;
    });
```

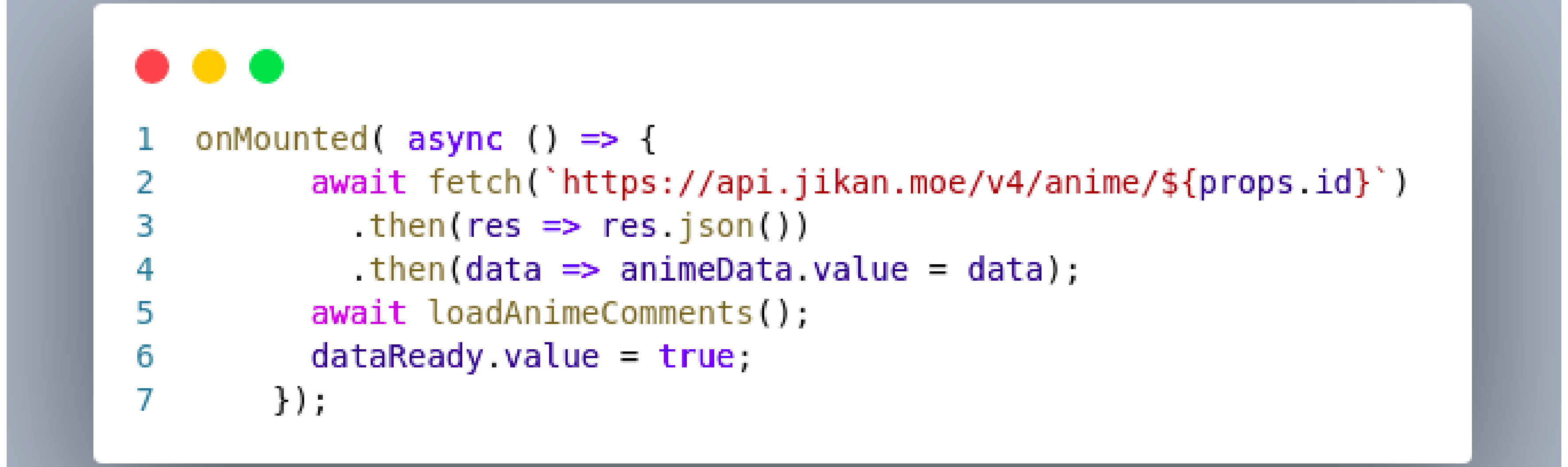


```
onMounted( async () => {
      await fetch(`https://api.jikan.moe/v4/anime/${props.id}`)
        .then(res => res.json())
        .then(data => animeData.value = data);
      await loadAnimeComments();
      dataReady.value = true;
    });
```

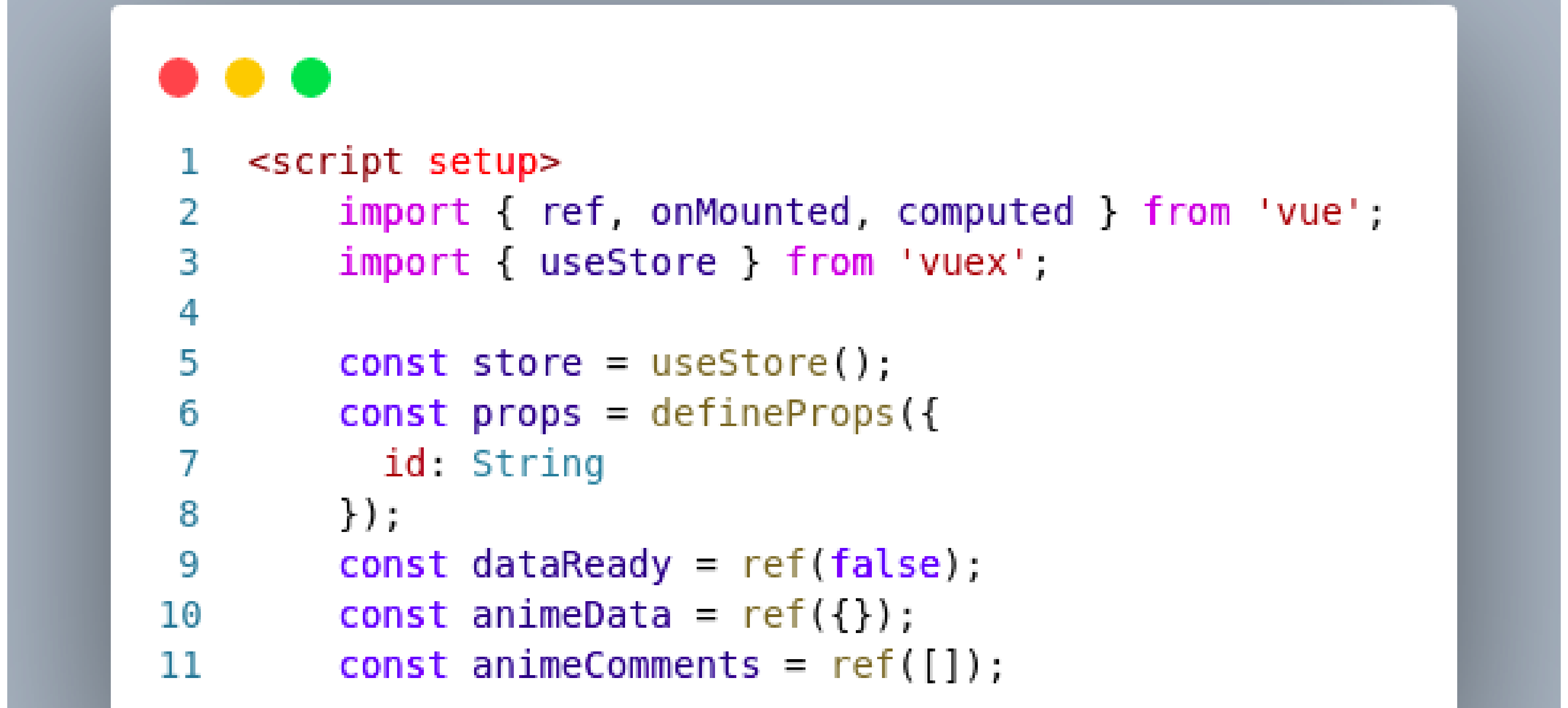


```
<script setup>
    import { ref, onMounted, computed } from 'vue';
    import { useStore } from 'vuex';

    const store = useStore();
    const props = defineProps({
      id: String
    });
    const dataReady = ref(false);
    const animeData = ref({});
    const animeComments = ref([]);
```

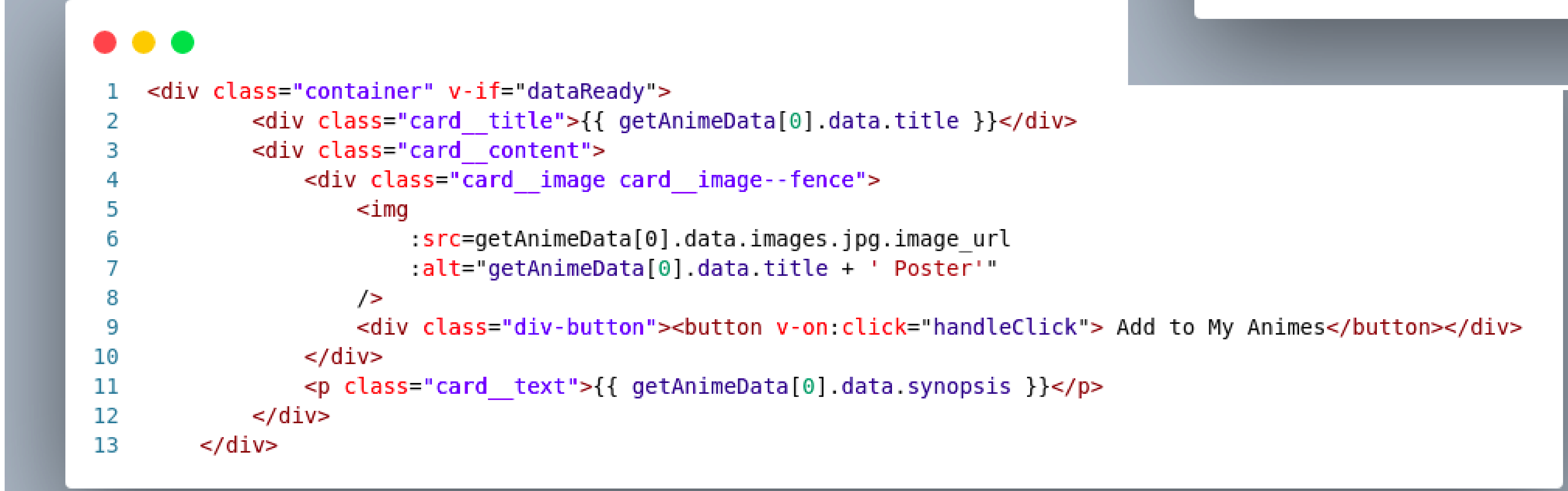


```
onMounted( async () => {
      await fetch(`https://api.jikan.moe/v4/anime/${props.id}`)
        .then(res => res.json())
        .then(data => animeData.value = data);
      await loadAnimeComments();
      dataReady.value = true;
    });
```



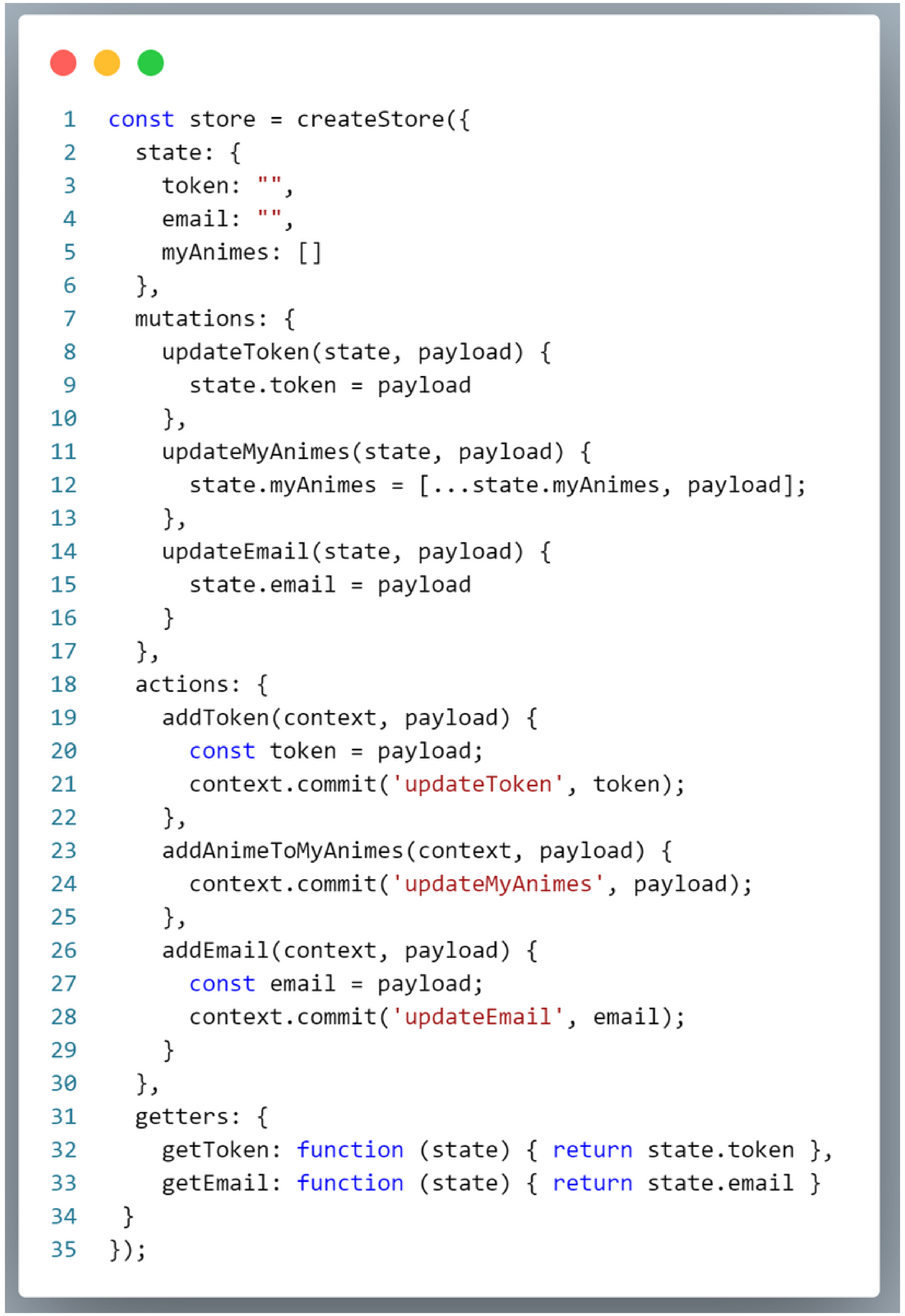
```
<script setup>
    import { ref, onMounted, computed } from 'vue';
    import { useStore } from 'vuex';

    const store = useStore();
    const props = defineProps({
      id: String
    });
    const dataReady = ref(false);
    const animeData = ref({});
    const animeComments = ref([]);
```

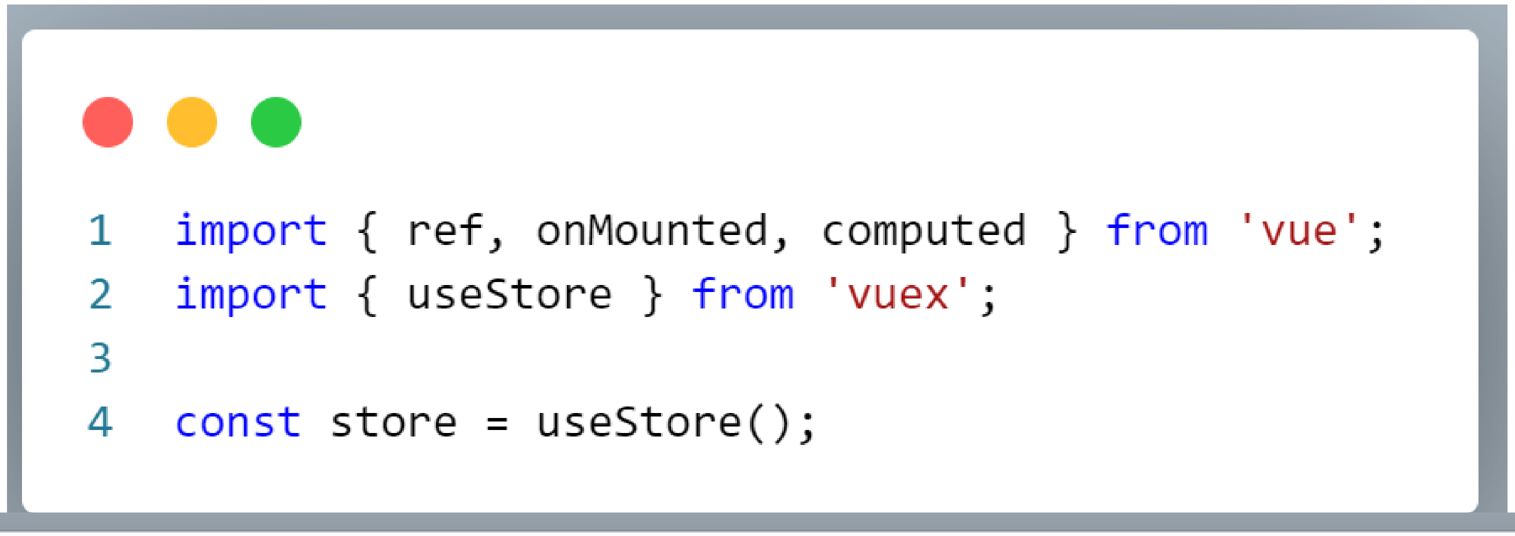


```
<div class="container" v-if="dataReady">
        <div class="card__title">{{ getAnimeData[0].data.title }}</div>
        <div class="card__content">
            <div class="card__image card__image--fence">
                <img
                    :src=getAnimeData[0].data.images.jpg.image_url
                    :alt="getAnimeData[0].data.title + ' Poster'"
                />
                <div class="div-button"><button v-on:click="handleClick"> Add to My Animes</button></div>
            </div>
            <p class="card__text">{{ getAnimeData[0].data.synopsis }}</p>
        </div>
    </div>
```

VUEX

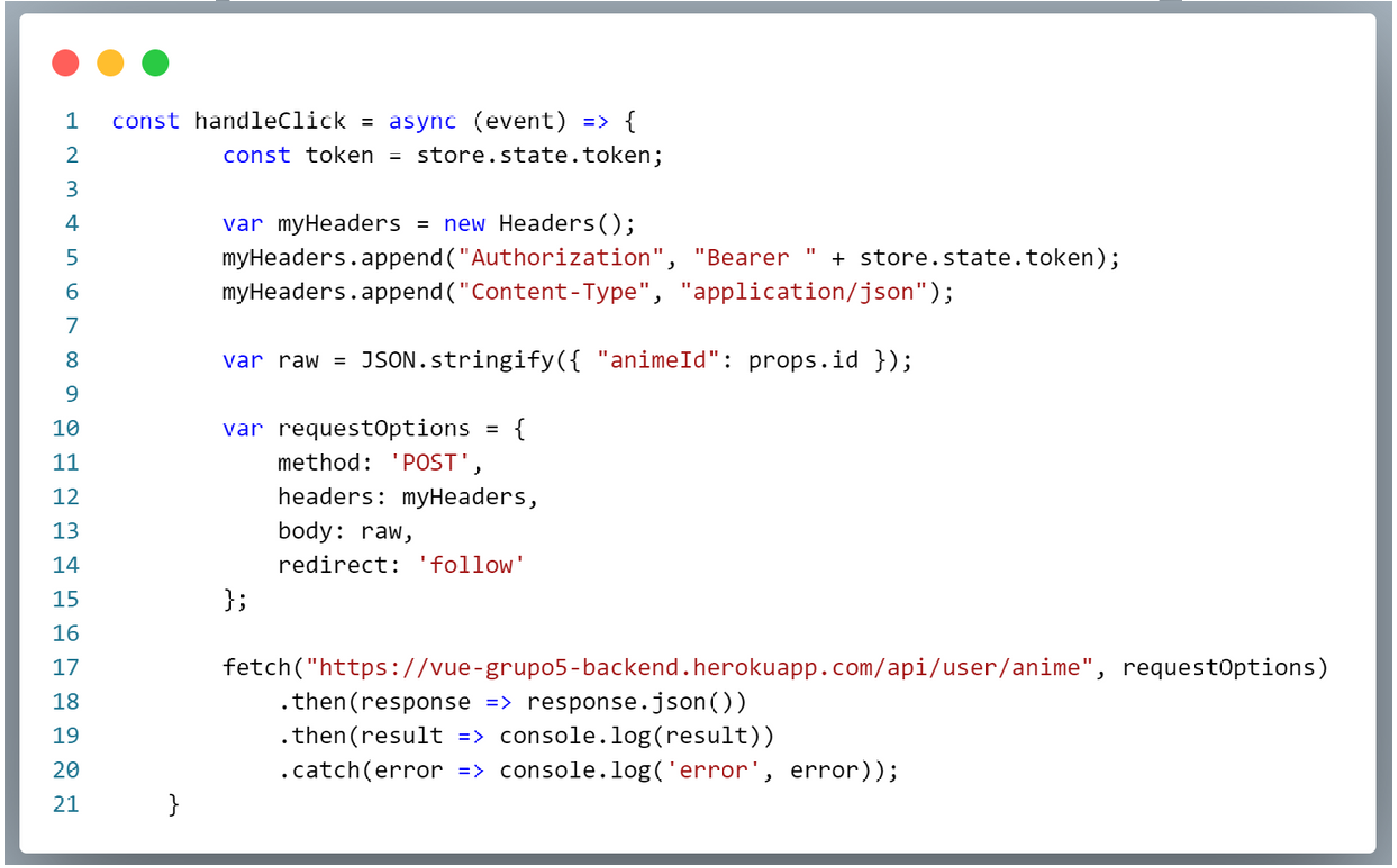


```
const store = createStore({
  state: {
    token: "",
    email: "",
    myAnimes: []
  },
  mutations: {
    updateToken(state, payload) {
      state.token = payload
    },
    updateMyAnimes(state, payload) {
      state.myAnimes = [...state.myAnimes, payload];
    },
    updateEmail(state, payload) {
      state.email = payload
    }
  },
  actions: {
    addToken(context, payload) {
      const token = payload;
      context.commit('updateToken', token);
    },
    addAnimeToMyAnimes(context, payload) {
      context.commit('updateMyAnimes', payload);
    },
    addEmail(context, payload) {
      const email = payload;
      context.commit('updateEmail', email);
    }
  },
  getters: {
    getToken: function (state) { return state.token },
    getEmail: function (state) { return state.email }
 }
});
```



```
import { ref, onMounted, computed } from 'vue';
import { useStore } from 'vuex';

const store = useStore();
```



```
const handleClick = async (event) => {
        const token = store.state.token;

        var myHeaders = new Headers();
        myHeaders.append("Authorization", "Bearer " + store.state.token);
        myHeaders.append("Content-Type", "application/json");

        var raw = JSON.stringify({ "animeId": props.id });

        var requestOptions = {
            method: 'POST',
            headers: myHeaders,
            body: raw,
            redirect: 'follow'
        };

        fetch("https://vue-grupo5-backend.herokuapp.com/api/user/anime", requestOptions)
            .then(response => response.json())
            .then(result => console.log(result))
            .catch(error => console.log('error', error));
    }
```

DIFICULTADES Y CONCLUSIONES

Simpleza en la construcción de componentes y scripts

Pocas referencias para solucionar problemas a diferencia de React

Interesante herramienta para desarrollo frontend

Pueden generarse archivos muy grandes en cada componente

Fácil curva de aprendizaje y elegante

T5: Vue.js

Javiera Inostroza, Elías Sabja, Samuel Zúñiga