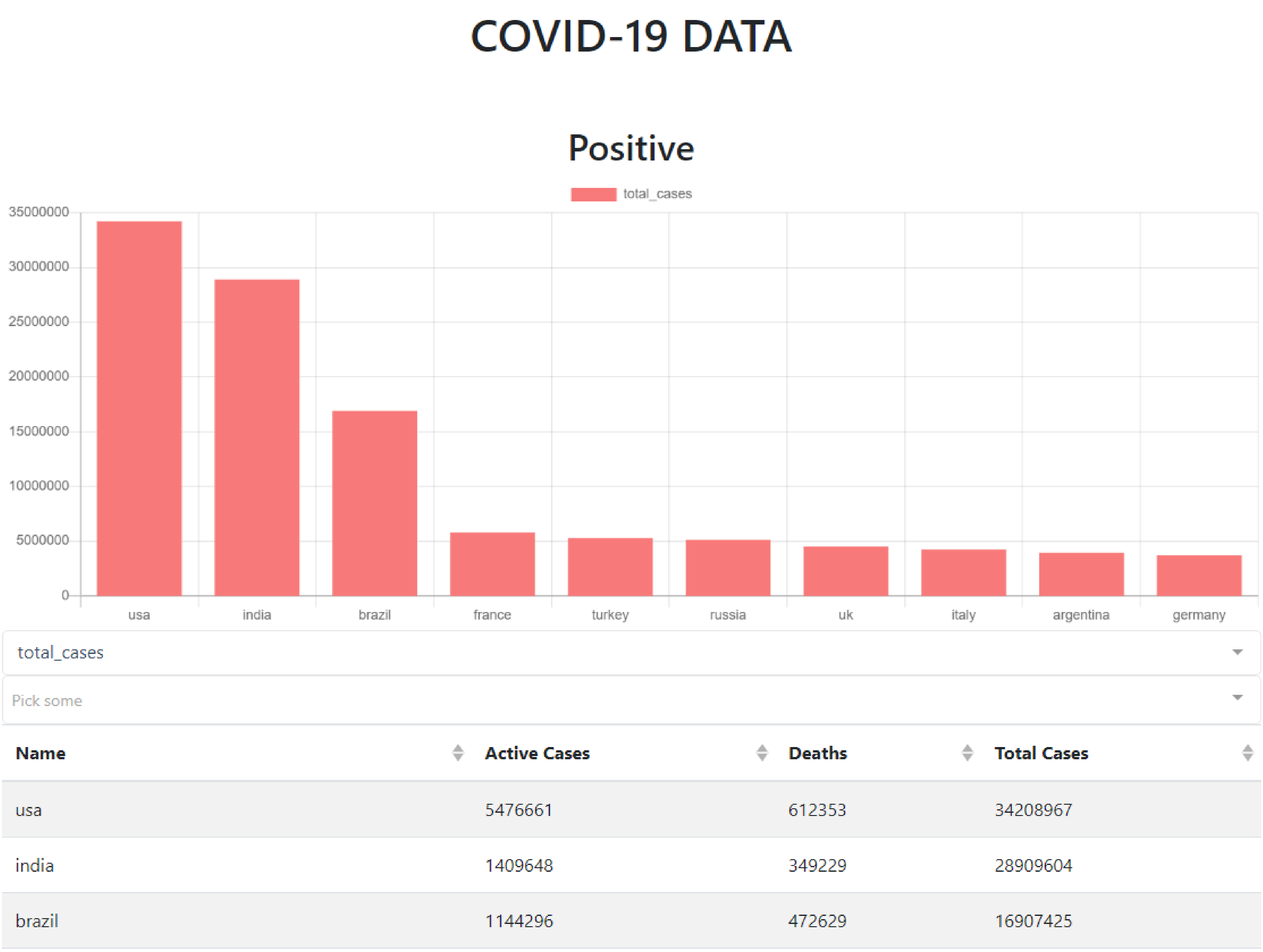
VUE.JS
+
VUEX

DEMO

Objetivo



total_cases

Pick some

Name	Active Cases	Deaths	Total Cases
usa	5476661	612353	34208967
india	1409648	349229	28909604
brazil	1144296	472629	16907425

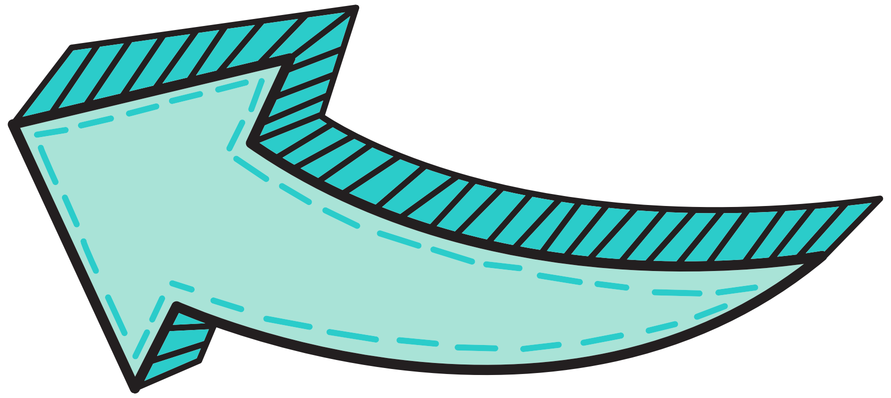
Construir una aplicación en base al framework JavaScript Vue.js

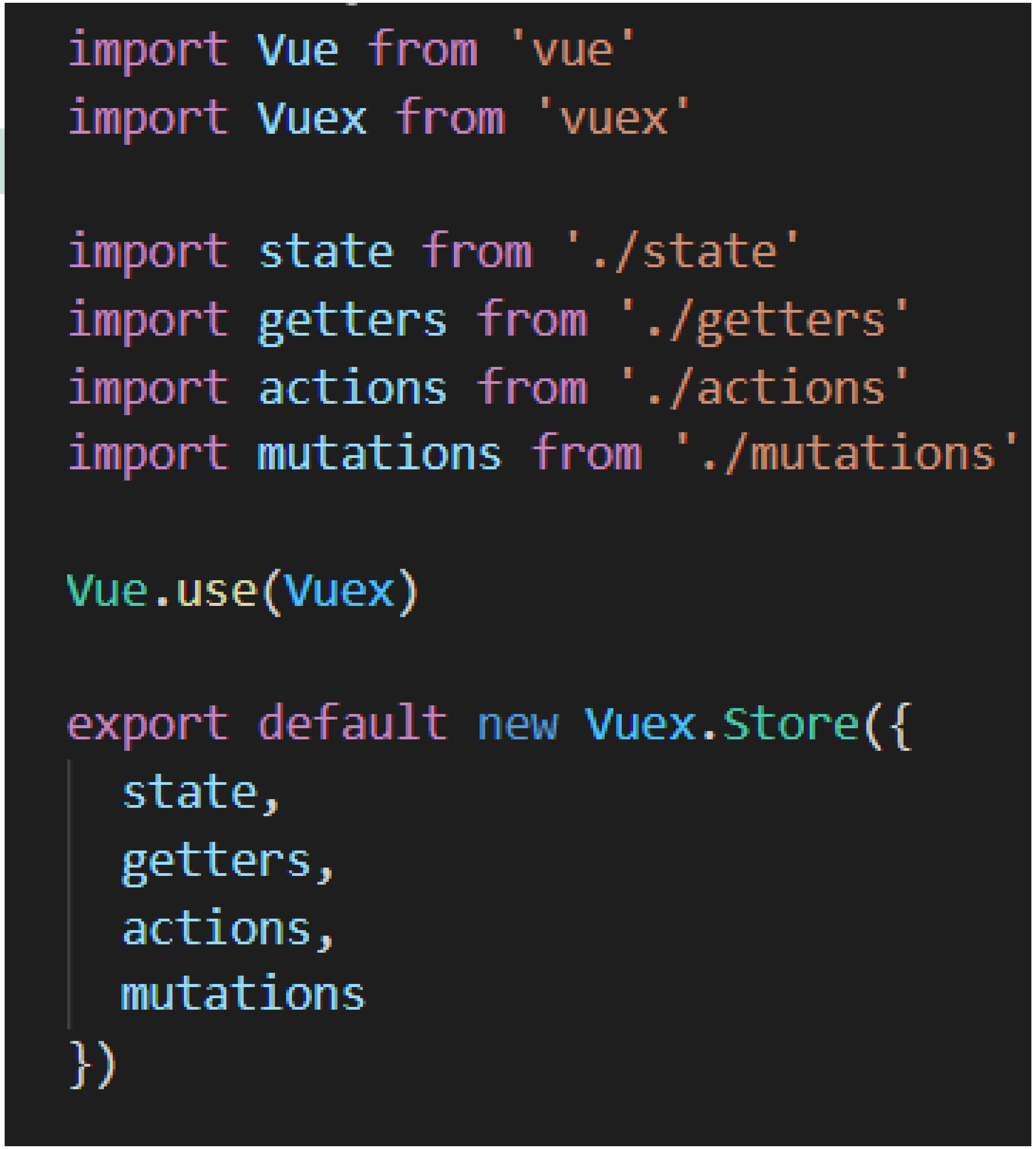
En nuestro caso, generamos una aplicación de tracking de información relacionada a COVID-19

Vuex

Se definen:

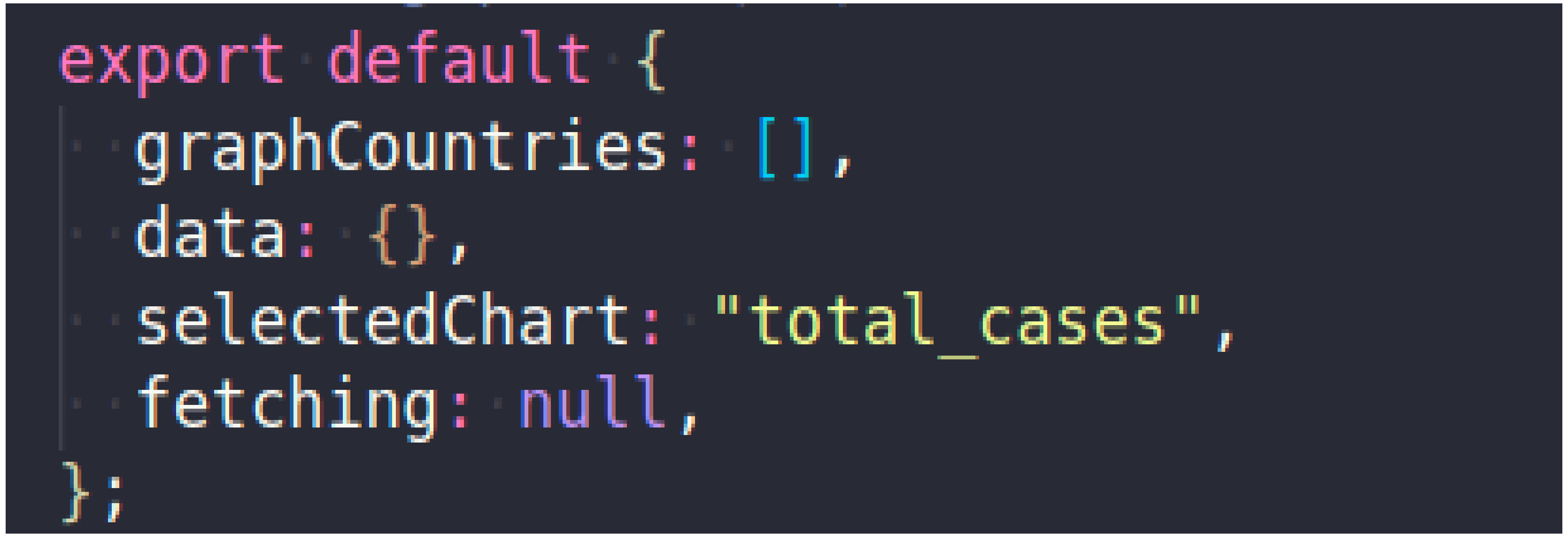
- state
- getters
- actions
- mutations





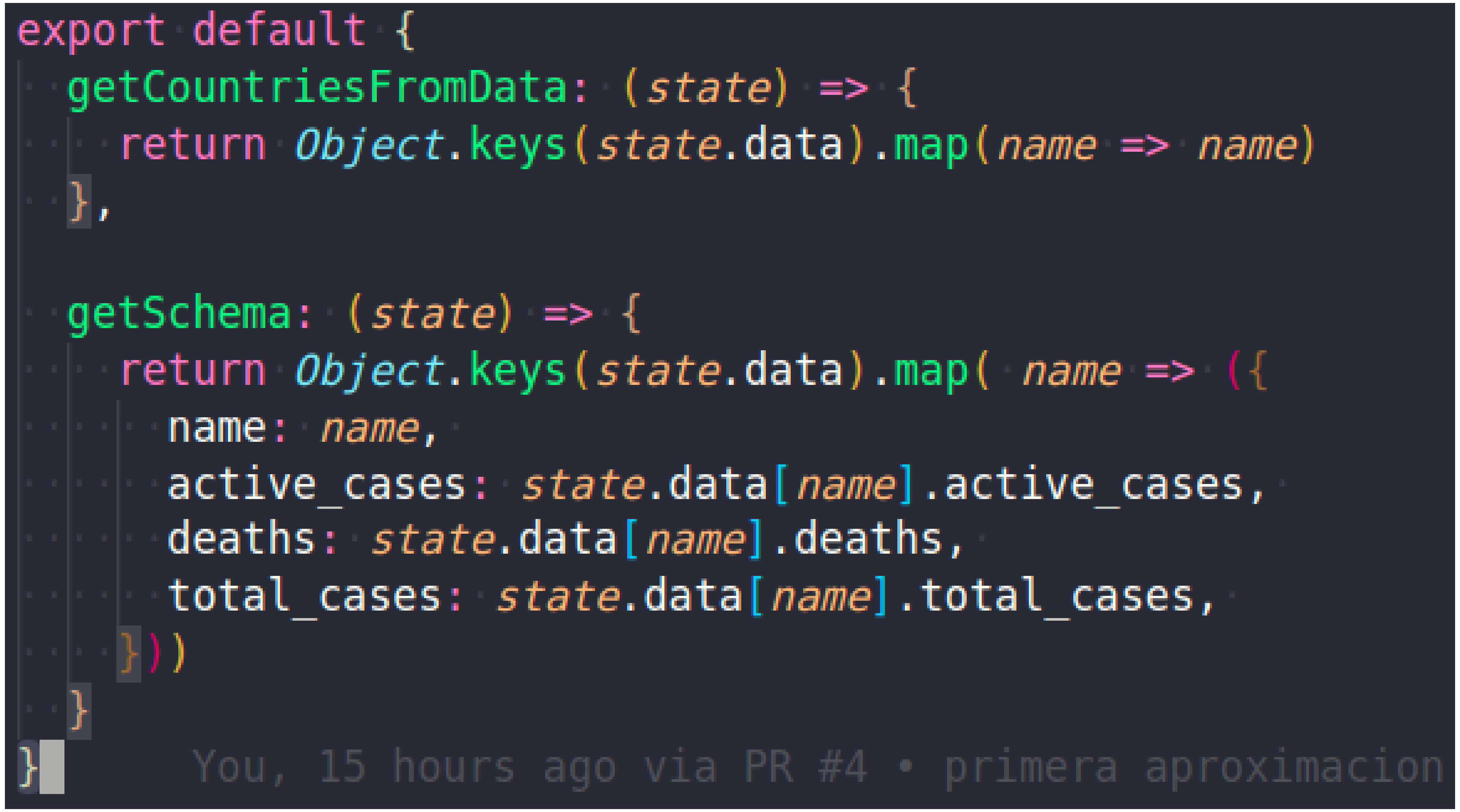
```
import Vue from 'vue'
import Vuex from 'vuex'

import state from './state'
import getters from './getters'
import actions from './actions'
import mutations from './mutations'

Vue.use(Vuex)

export default new Vuex.Store({
  state,
  getters,
  actions,
  mutations
})
```

Vuex - state



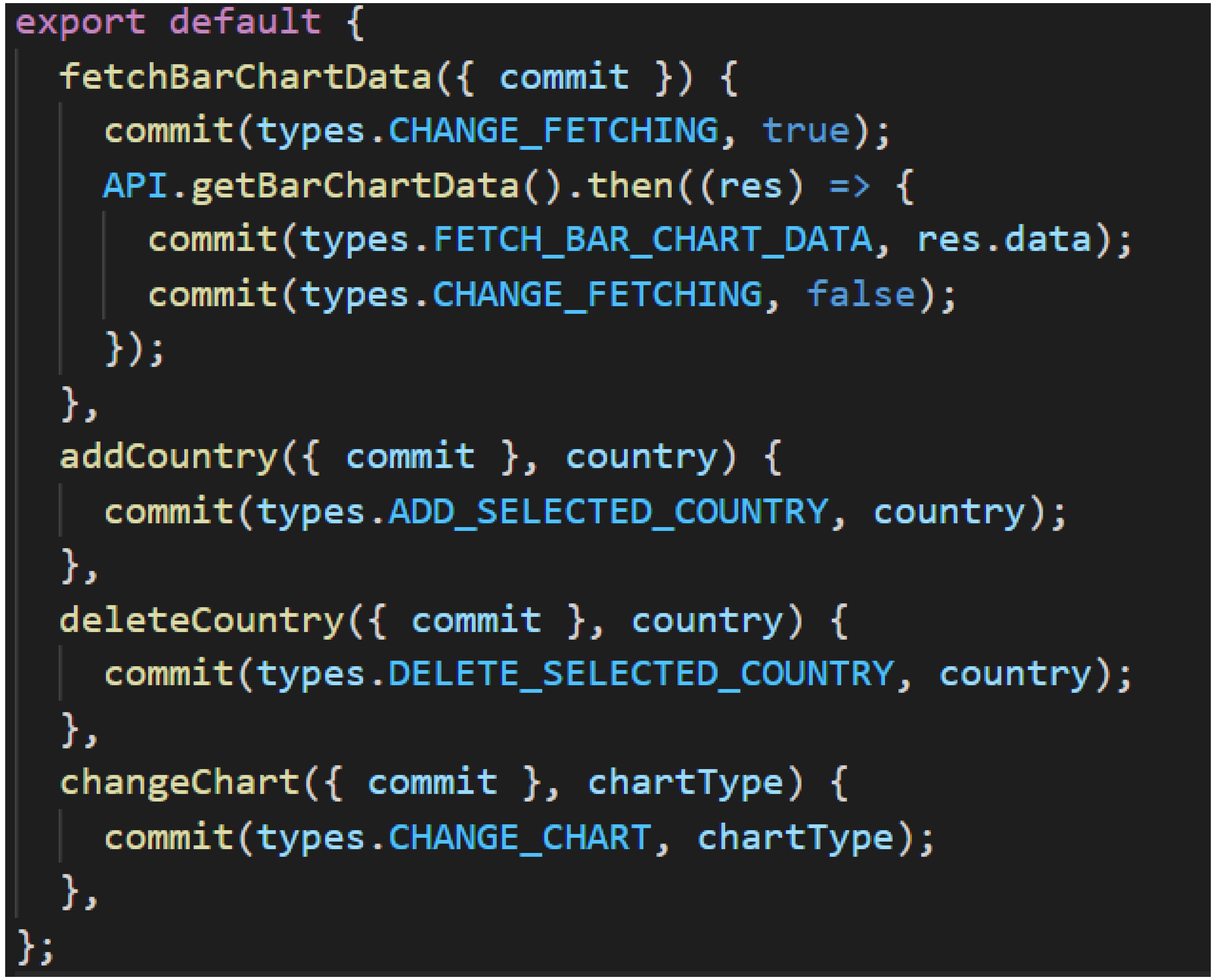
```
export default {
  graphCountries: [],
  data: {},
  selectedChart: "total_cases",
  fetching: null,
};
```

Vuex - getters



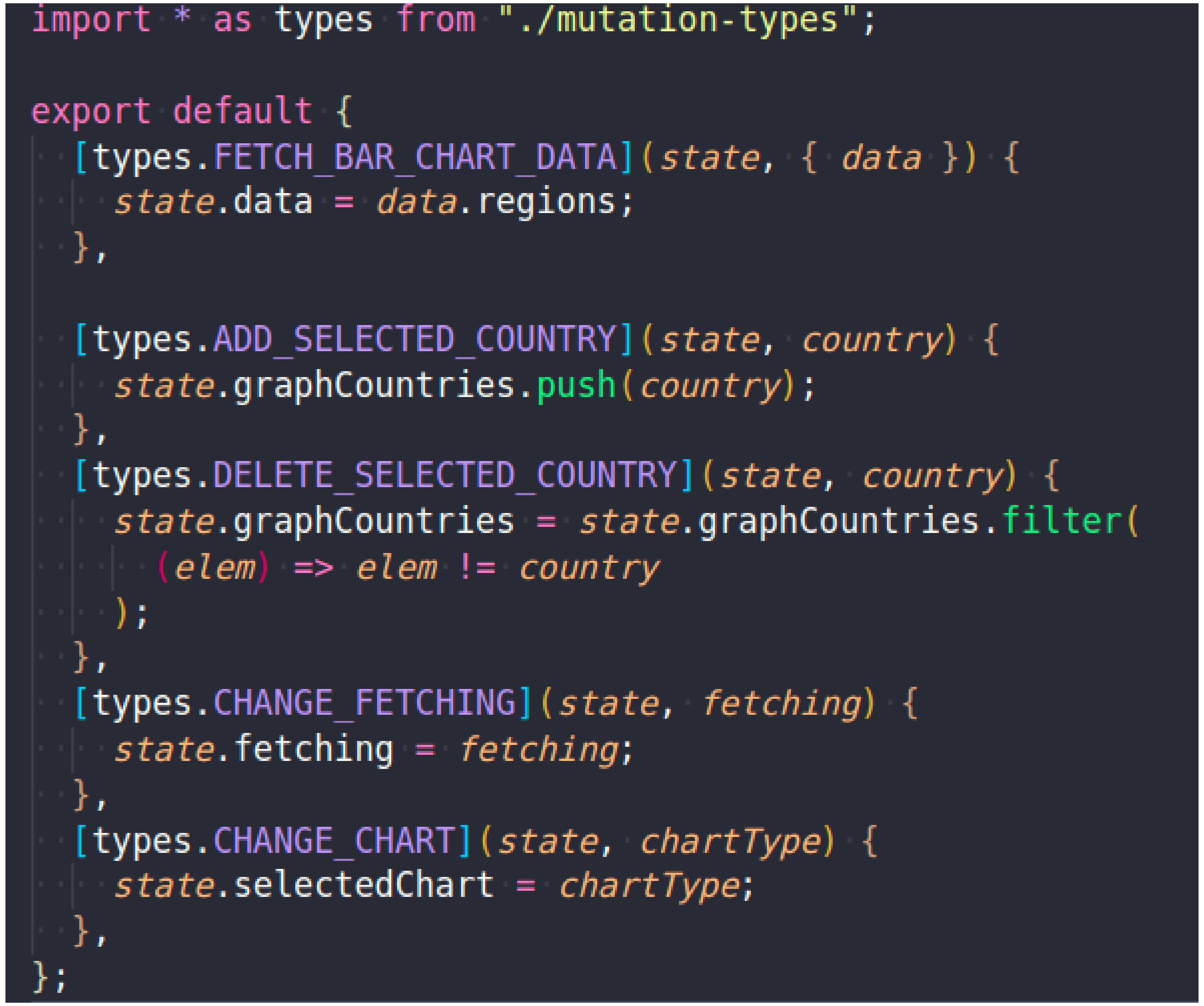
```
export default {
  getCountriesFromData: (state) => {
    return Object.keys(state.data).map(name => name)
  },

  getSchema: (state) => {
    return Object.keys(state.data).map( name => ({
      name: name,
      active_cases: state.data[name].active_cases,
      deaths: state.data[name].deaths,
      total_cases: state.data[name].total_cases,
    }))
  }
}      You, 15 hours ago via PR #4 • primera aproximacion
```

Vuex - actions



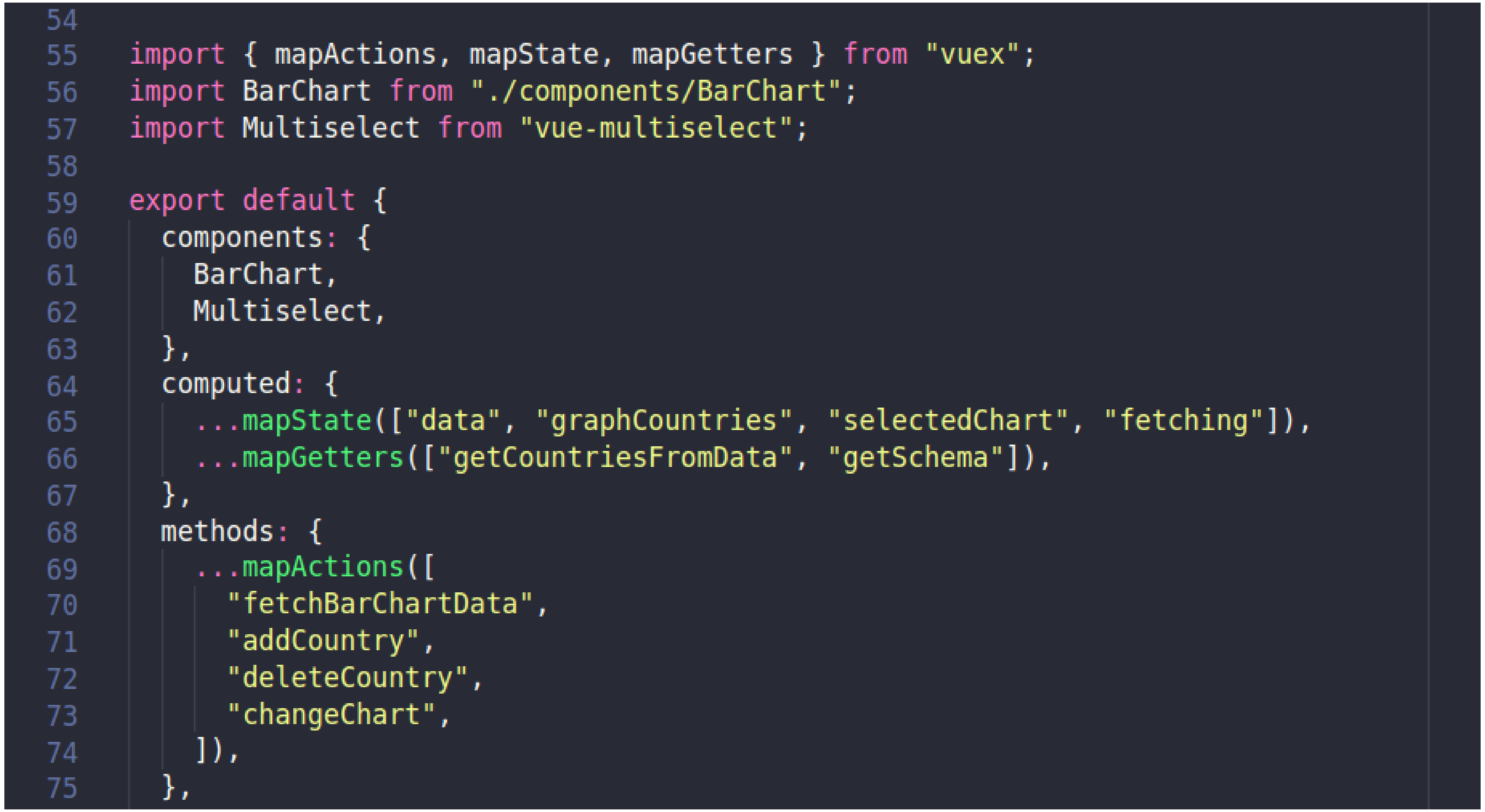
```
export default {
  fetchBarChartData({ commit }) {
    commit(types.CHANGE_FETCHING, true);
    API.getBarChartData().then((res) => {
      commit(types.FETCH_BAR_CHART_DATA, res.data);
      commit(types.CHANGE_FETCHING, false);
    });
  },
  addCountry({ commit }, country) {
    commit(types.ADD_SELECTED_COUNTRY, country);
  },
  deleteCountry({ commit }, country) {
    commit(types.DELETE_SELECTED_COUNTRY, country);
  },
  changeChart({ commit }, chartType) {
    commit(types.CHANGE_CHART, chartType);
  },
};
```

Vuex - mutations



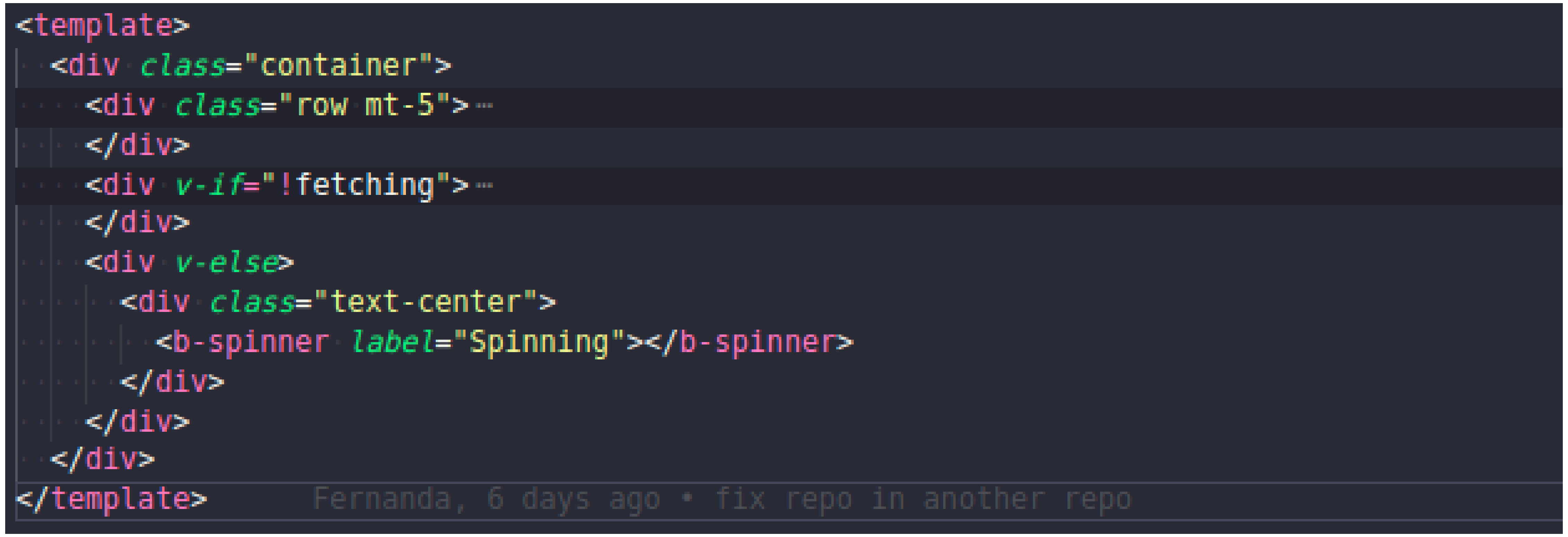
```
import * as types from "./mutation-types";

export default {
  [types.FETCH_BAR_CHART_DATA](state, { data }) {
    state.data = data.regions;
  },

  [types.ADD_SELECTED_COUNTRY](state, country) {
    state.graphCountries.push(country);
  },
  [types.DELETE_SELECTED_COUNTRY](state, country) {
    state.graphCountries = state.graphCountries.filter(
      (elem) => elem != country
    );
  },
  [types.CHANGE_FETCHING](state, fetching) {
    state.fetching = fetching;
  },
  [types.CHANGE_CHART](state, chartType) {
    state.selectedChart = chartType;
  },
};
```

App.vue



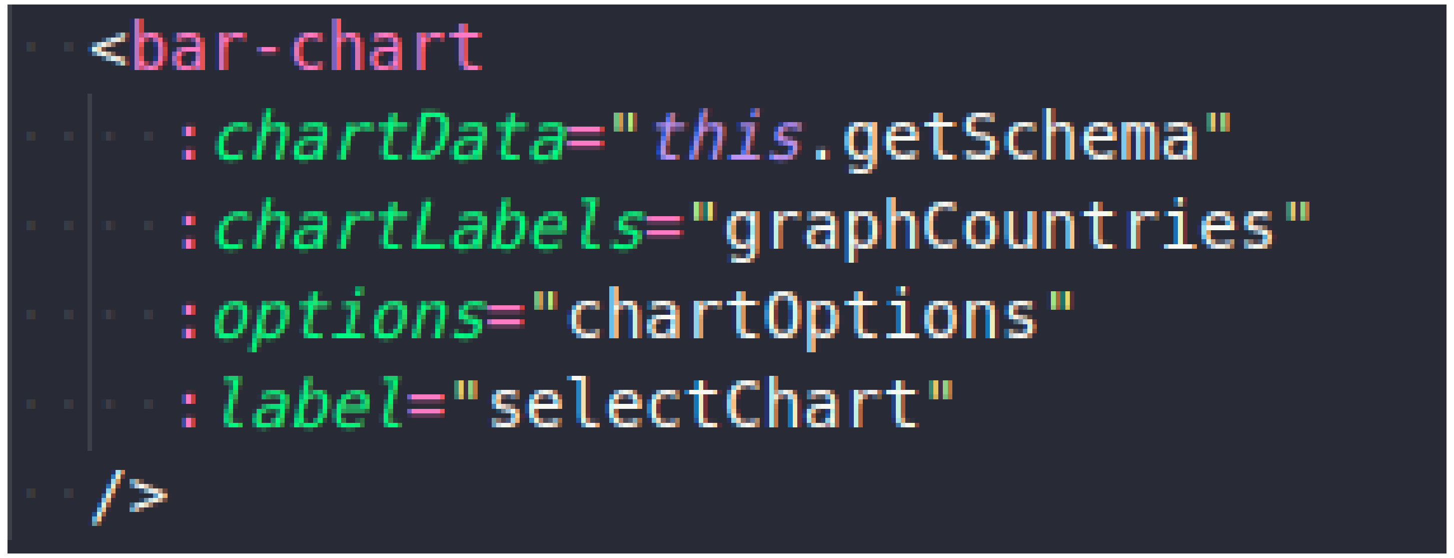
```
import { mapActions, mapState, mapGetters } from "vuex";
import BarChart from "./components/BarChart";
import Multiselect from "vue-multiselect";

export default {
  components: {
    BarChart,
    Multiselect,
  },
  computed: {
    ...mapState(["data", "graphCountries", "selectedChart", "fetching"]),
    ...mapGetters(["getCountriesFromData", "getSchema"]),
  },
  methods: {
    ...mapActions([
      "fetchBarChartData",
      "addCountry",
      "deleteCountry",
      "changeChart",
    ]),
  },
```

App.vue

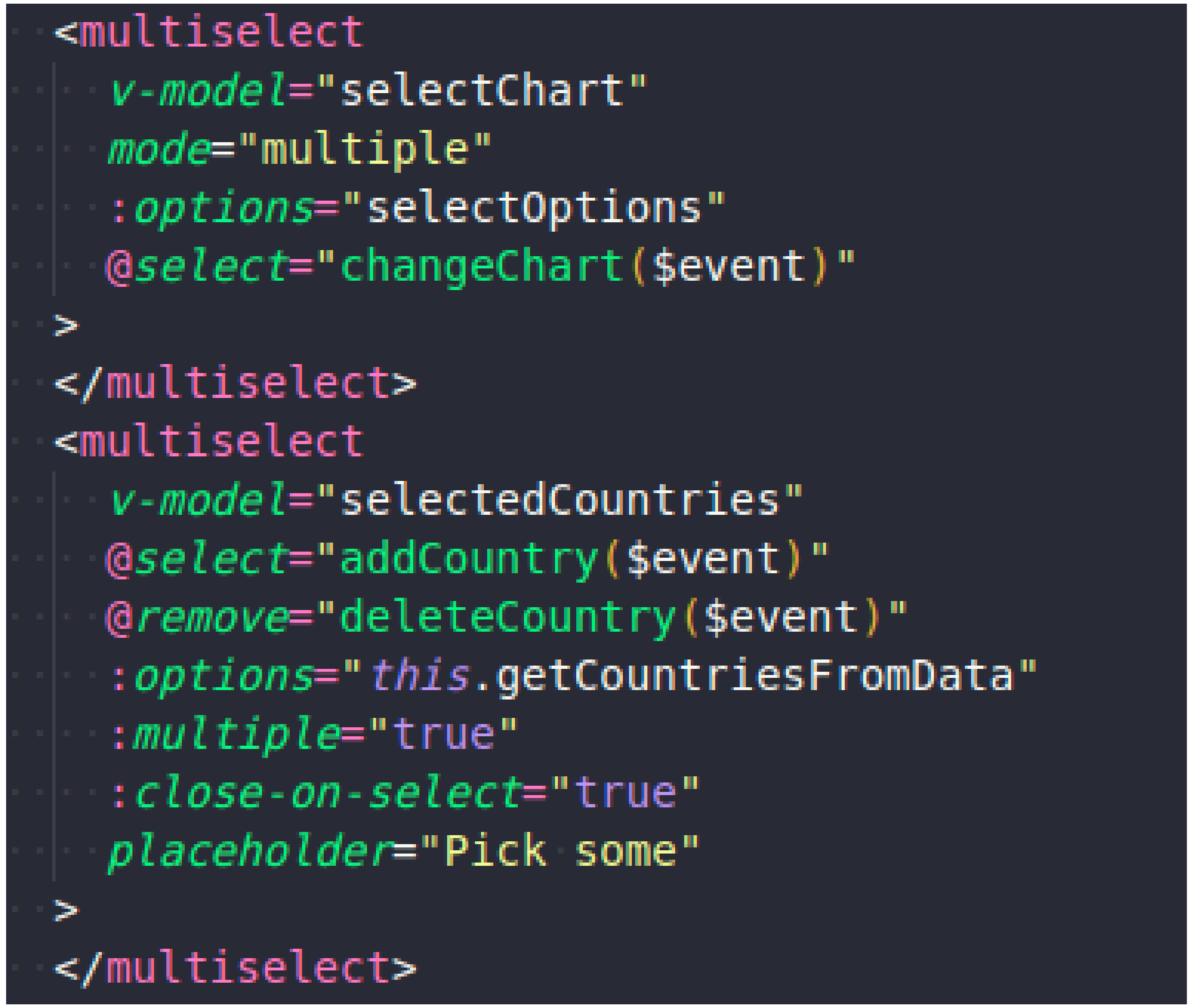


```
<template>
  <div class="container">
    <div class="row mt-5">⋯
    </div>
    <div v-if="!fetching">⋯
    </div>
    <div v-else>
      <div class="text-center">
        <b-spinner label="Spinning"></b-spinner>
      </div>
    </div>
  </div>
</template>      Fernanda, 6 days ago • fix repo in another repo
```

App.vue

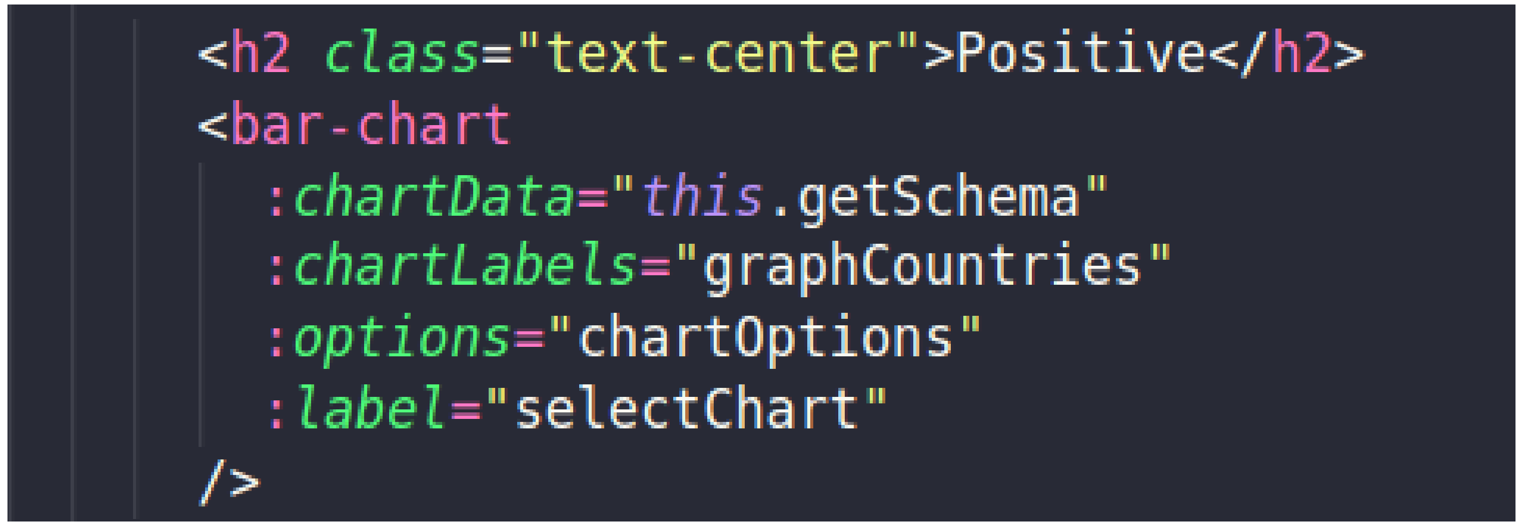


```
<bar-chart
    :chartData="this.getSchema"
    :chartLabels="graphCountries"
    :options="chartOptions"
    :label="selectChart"
/>
```

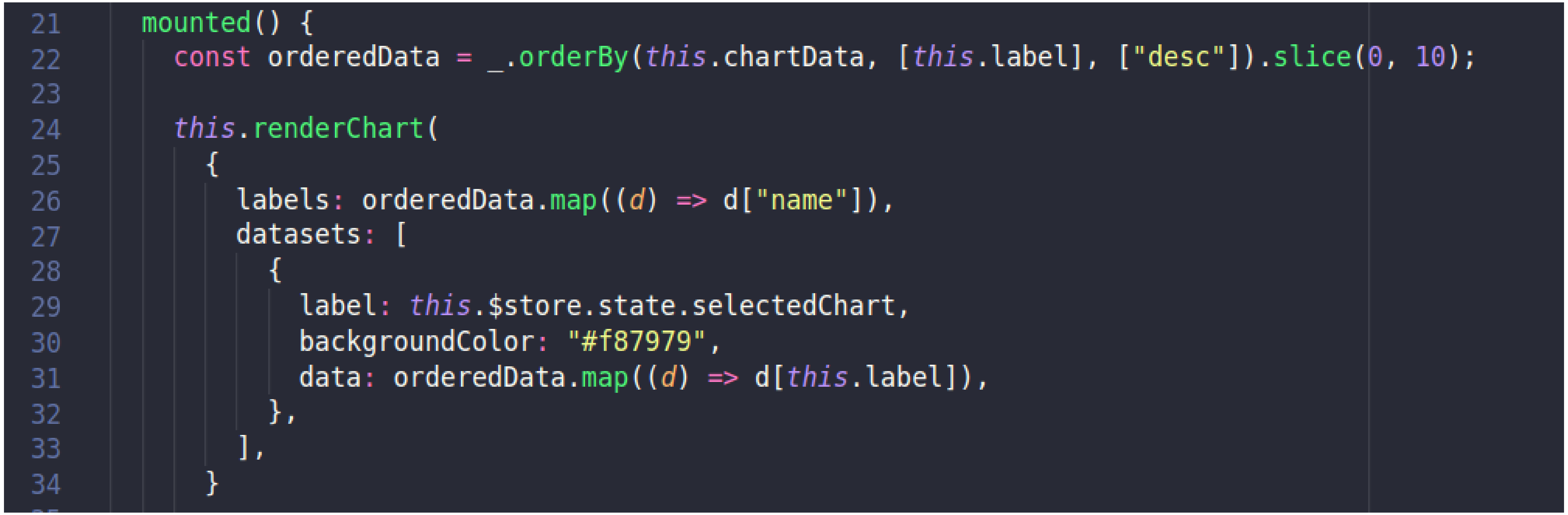


```
<multiselect
    v-model="selectChart"
    mode="multiple"
    :options="selectOptions"
    @select="changeChart($event)"
>
</multiselect>
<multiselect
    v-model="selectedCountries"
    @select="addCountry($event)"
    @remove="deleteCountry($event)"
    :options="this.getCountriesFromData"
    :multiple="true"
    :close-on-select="true"
    placeholder="Pick some"
>
</multiselect>
```

Vue - chart

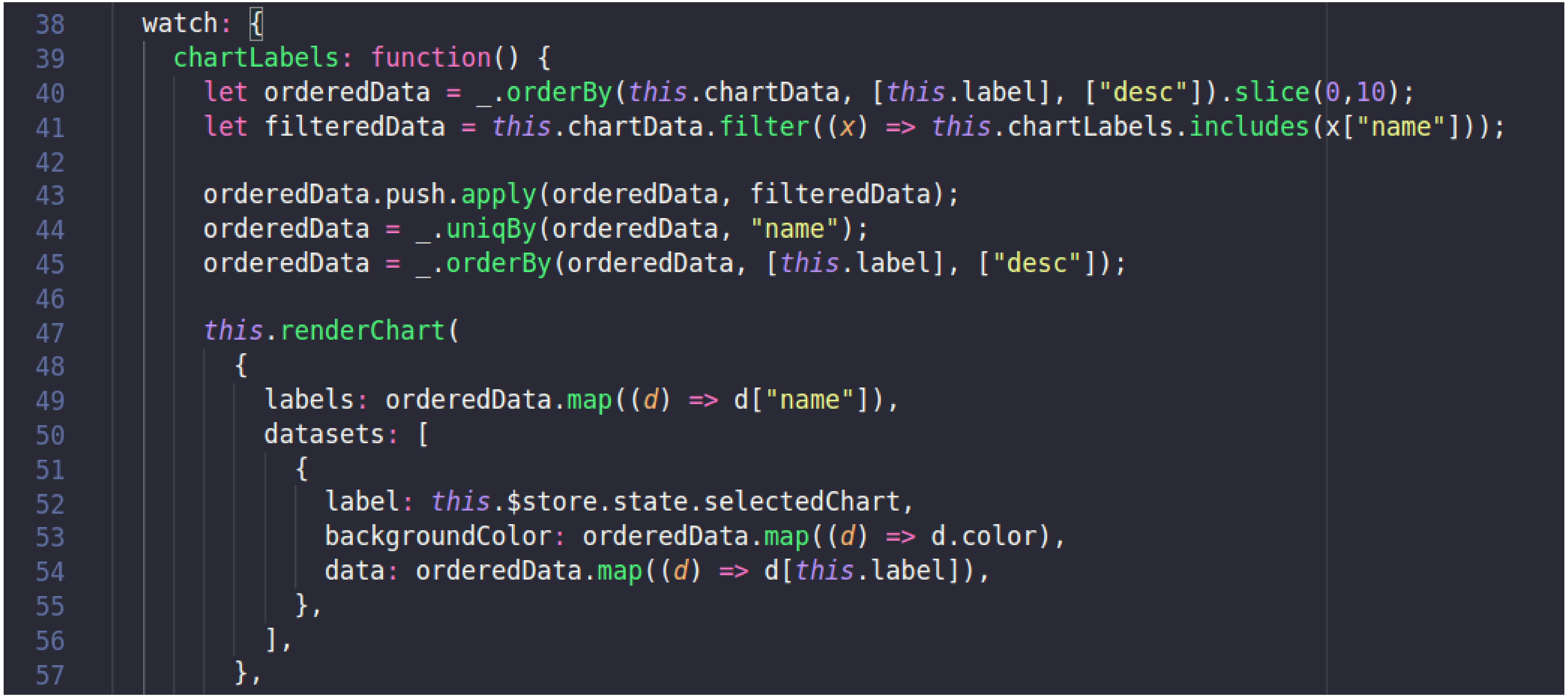


```
<h2 class="text-center">Positive</h2>
<bar-chart
  :chartData="this.getSchema"
  :chartLabels="graphCountries"
  :options="chartOptions"
  :label="selectChart"
/>
```



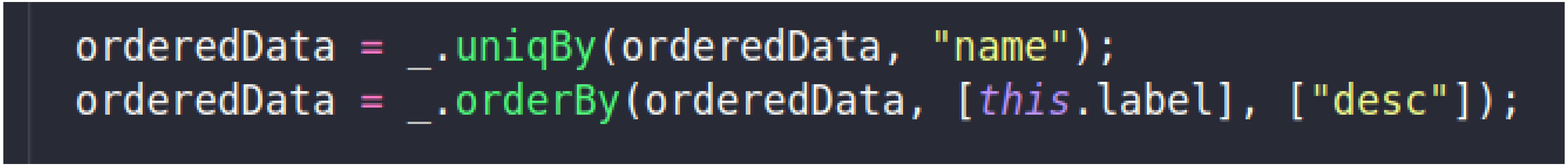
```
mounted() {
  const orderedData = _.orderBy(this.chartData, [this.label], ["desc"]).slice(0, 10);

  this.renderChart(
    {
      labels: orderedData.map((d) => d["name"]),
      datasets: [
        {
          label: this.$store.state.selectedChart,
          backgroundColor: "#f87979",
          data: orderedData.map((d) => d[this.label]),
        },
      ],
    }
```

Vue - chart



```
watch: {
  chartLabels: function() {
    let orderedData = _.orderBy(this.chartData, [this.label], ["desc"]).slice(0,10);
    let filteredData = this.chartData.filter((x) => this.chartLabels.includes(x["name"]));

    orderedData.push.apply(orderedData, filteredData);
    orderedData = _.uniqBy(orderedData, "name");
    orderedData = _.orderBy(orderedData, [this.label], ["desc"]);

    this.renderChart(
      {
        labels: orderedData.map((d) => d["name"]),
        datasets: [
          {
            label: this.$store.state.selectedChart,
            backgroundColor: orderedData.map((d) => d.color),
            data: orderedData.map((d) => d[this.label]),
          },
        ],
      },
```

Lodash



```
orderedData = _.uniqBy(orderedData, "name");
orderedData = _.orderBy(orderedData, [this.label], ["desc"]);
```

MUCHAS GRACIAS