Store/index.js

import { createStore } from 'vuex'

import axios from 'axios';

export default createStore({

  state: {

    posts: [],

    // 1111111111111111111111

    dialogVisible: false,

    isPostLoading: false, // индикатор загрузки - может заменить на крутилку?

    selectedSort: '',

    searchQuery: '',

    page: 1,

    limit: 10,

    totalPages: 0,

    sortOptions: [

        {value: 'title', name: 'По названию'},

        {value: 'body', name: 'По содержанию'},

    // {value: 'id', name: 'По дате'},   // добавить сортировку по дате?

    ]

  },

  getters: {

    // computed

    // используем эту ф-ю как переменную, земенив :posts="posts" в PostList на :posts="sortedPosts"

    sortedPosts(state) {

      return [...state.posts].sort((post1, post2) => post1[state.selectedSort]?.localeCompare(post2[state.selectedSort]))

    },

    sortedAndSearhedPosts(state, getters) {

        return getters.sortedPosts.filter(post => post.title.toLowerCase().includes(state.searchQuery.toLowerCase()))

    }

  },

  mutations: {

    // для изменения состояния

    setPosts(state, posts) {

      state.posts = posts;

    },

    // 11111111111

    dialogVisible(state, bool) {

      state.dialogVisible = bool

    },

    setLoading(state, bool) {

      state.isPostLoading = bool

    },

    setPage(state, page) {

        state.page = page

    },

    setSelectedSort(state, selectedSort) {

        state.selectedSort = selectedSort

    },

    setTotalPages(state, totalPages) {

        state.totalPages = totalPages

    },

    setSearchQuery(state, searchQuery) {

        state.searchQuery = searchQuery

    },

    // 1111111111111111111111

    createPost(state ,post) {

      state.posts.unshift(post);

      state.dialogVisible = false;

  },

  },

  actions: {

    // https://ru.vuejs.org/v2/cookbook/using-axios-to-consume-apis.html

    async fetchPosts({state, commit}) {

      try {

        commit('setLoading', true);

        // this.$refs.observer

        // делаем запрос на сервис

        const response = await axios.get('https://jsonplaceholder.typicode.com/posts', {

          params: {

            \_page: state.page,

            \_limit: state.limit

          }

        });

        // высчитываем ск-ко страниц. Общее кол-во постов делим на кол-во постов на 1 странице, округляя в большую сторону (101 / 10 = 11)

        commit('setTotalPages', Math.ceil(response.headers['x-total-count'] / state.limit));

        commit('setPosts', response.data);

      } catch (error) {

        // alert('Ошибка');

        console.log(error);

      } finally {

        commit('setLoading', false);

      }

    },

    // подгрузка свежих постов

    async loadMorePosts({state, commit}) {

      try {

        commit('setPage', state.page + 1);  //////////////////////////////// +=1

        // this.isPostLoading = true;

        // делаем запрос на сервис

        const response = await axios.get('https://jsonplaceholder.typicode.com/posts', {

          params: {

            \_page: state.page,

            \_limit: state.limit

          }

        });

        // высчитываем ск-ко страниц. Общее кол-во постов делим на кол-во постов на 1 странице, округляя в большую сторону (101 / 10 = 11)

        commit('setTotalPages', Math.ceil(response.headers['x-total-count'] / state.limit));

        commit('setPosts', [...state.posts, ...response.data]);

      } catch (error) {

          // alert('Ошибка');

          console.log(error);

      }

    },

  },

  modules: {

  }

})