\*\*\* npm install --save vuex

Step 1 : src er under e store->store.js create kora holo

\*\*\* store.js e jevabe shob normally korte hobe,

//store.js,

import Vue from "vue";

import Vuex from "vuex";

Vue.use(Vuex);

export const store = new Vuex.Store({

  state: {

    counter: 0,

  },

});

//now in main.js (importing store),

import Vue from "vue";

import App from "./App.vue";

import { store } from "./store/store.js";

new Vue({

  el: "#app",

  store,

  render: (h) => h(App),

});

\*\*\*state e access normally vue3 er motoi, this.$store.state.counter;

\*\*\*store.state er value computed e thakbe, eta ekta template example,

<template>

  <p>Counter is: {{ counter }}</p>

</template>

<script>

export default {

  computed: {

    counter() {

      return this.$store.state.counter;

    },

  },

};

</script>

\*\*\*fix for using …map functions in vue v2, etuk fix na korle spread operator kaj korbe na vue v2 te,



npm install --save webpack-preset webpack-preset-babel webpack-preset-babel-stage-2

//in babelrc,

{

  "presets": [["es2015", { "modules": false }], ["stage-2"]]

}

\*\*\* Two way binding in state management with commit(mutation) example,

<template>

  <div>

    <input type="text" :value="value" @input="updateValue" />

  </div>

</template>

<script>

export default {

  methods: {

    updateValue(value) {

      this.$store.commit("updateValue", event.target.value);

    },

  },

};

</script>

\*\*\*Modules

Prothome ekta modules folder banate hobe ebong modules->counter.js banana holo, then counter.js e code jevabe likhte hobe,

const state = {

  counter: 0,

};

export default { //exporting

  state,

};

Then states er moddhe jevabe modules add korte hobe,

import Vue from "vue";

import Vuex from "vuex";

import counter from "./modules/counter.js";

Vue.use(Vuex);

export const store = new Vuex.Store({

  state: {},

  getters: {

    getCounter(state) {

      return state.counter.counter;

    },

  },

  mutations: {

    updateValue(state, value) {

      state.counter.counter = value;

    },

  },

  modules: {

    counter: counter, //modules

  },

});

\*\*modules theke asha value gula object akare ashe , example { counter : 0 }, tai counter.counter lekha hoise

\*\*\*\* import \* as actions from ‘./…..’; //eta hocche shob import korte pare

\*\*\*\*\*Namespace to avoid naming problems

import Vue from "vue";

import Vuex from "vuex";

import counter from "./modules/counter.js";

import \* as types from "./types.js";

Vue.use(Vuex);

export const store = new Vuex.Store({

  state: {},

  getters: {

    getCounter(state) { //etake naming korte hobe ///////////////////////////

      return state.counter.counter;

    },

  },

  mutations: {

    updateValue(state, value) {

      state.counter.counter = value;

    },

  },

  modules: {

    counter: counter, //modules

  },

});

Step 1 : Prothome types.js name e ekta js file create kora holo,

export const GET\_COUNTER = "counter/GET\_COUNTER";

Step 2 :

evabe import kore nite hobe folder wise,

import \* as types from "./types.js";

Then evabe getters er moto nam set kore dite hobe,

//from state.js,

import Vue from "vue";

import Vuex from "vuex";

import counter from "./modules/counter.js";

import \* as types from "./types.js";

Vue.use(Vuex);

export const store = new Vuex.Store({

  state: {},

  getters: {

    [types.GET\_COUNTER](state) {

      //name change kora hoise ///////////////////////////

      return state.counter.counter;

    },

  },

  mutations: {

    updateValue(state, value) {

      state.counter.counter = value;

    },

  },

  modules: {

    counter: counter, //modules

  },

});

//now in any .vue file jekhane namespace kora nam access kora lagbe,

<template>

  <p>Counter is: {{ getCounter }}</p>

</template>

<script>

import { mapGetters } from "vuex";

import \* as types from "../types.js"; //importing type file

export default {

  computed: {

    ...mapGetters({ getCounter: types.GET\_COUNTER }), //types folder structure  and GET\_COUNTER holo set kora namespace

  },

};

</script>

types.GET\_COUNTER holo namespace kore asha state e set kora name, muloto evabei kore name spacing & getCounter e object use kora hoise karon getCounter ke as a property hishabe use korar jonno & types.GET\_COUNTER hocche namespace kora notun name.

\*\*\*\*\*\* import \* as something mane shob import hoye ashbe, onno vue file theke export default kora lagbe na cause import \* means shob e import kora hoise.

/////accessing value in App.vue example, ///////////

<template>

<base-container title="Vuex">

<h3>{{ $store.state.counter }}</h3>

<!-- evabe access kore $store er value -->

<button @click="count">Add 1</button>

</base-container>

</template>

<script>

import BaseContainer from './components/BaseContainer.vue';

export default {

components: {

BaseContainer

},

methods: {

count() {

this.$store.state.counter++;

}

}

};

</script>

<style>

\* {

box-sizing: border-box;

}

html {

font-family: sans-serif;

}

body {

margin: 0;

}

</style>

/////////////////mutation(setter) example ///////////////////////////////////////////////////////////////

main.js,

const store = createStore({

state() {

return {

counter: 0

};

},

mutations: {

increment(state) {

state.counter++;

}

}

});

in App.vue,

methods: {

addOne() {

this.$store.commit('increment');

}

}

//////////////////////mutation with parameters //////////////////////////////////////

App.vue,

methods: {

addOne() {

this.$store.commit('increment', 2);

}

}

main.js,

mutations: {

increment(state, v) {

state.counter = state.counter + v;

}

}

//////////////////////getters //////////////////////////////////////

//App.vue,

computed: {

counter() {

return this.$store.getters.finalCounter; //point it like a property

}

}

//main.js,

getters: {

finalCounter(state) {

return state.counter \* 2;

}

}

///////////getters with getter ///////////////////////////////////////////

//main.js,

getters: {

finalCounter(state) {

return state.counter \* 2;

},

doubleFinalCounter(state, getters) {

return state.counter + getters.finalCounter;

}

}

////////////////actions //////////////////////////////////////////////////////////////////////

//main.js,

actions: {

//lekhar system

increment(context, payload) {

//method name + context(targetted method) + payload

setTimeout(() => {

context.commit('increment', payload); //targetted method commit

}, 2000);

}

}

//App.vue,

methods: {

actionCount() {

return this.$store.dispatch({ type: 'increment', value: 10 });

}

}

///////////////////////////////mapGetters(Getters) and mapActions(Actions) ///////////////////////////////////

<template>

<div>

<h3>{{ doubleFinalCounter }}</h3>

<p>We do more ...</p>

<button @click="increment({ value: 10 })">Action Count</button>

</div>

</template>

<script>

import { mapGetters } from 'vuex';

import { mapActions } from 'vuex';

export default {

computed: {

...mapGetters(['doubleFinalCounter'])

// counter() {

// return this.$store.getters.doubleFinalCounter; //point it like a property

// }

},

methods: {

// actionCount() {

// return this.$store.dispatch({ type: 'increment', value: 10 });

// }

...mapActions(['increment']) //object o pass kora jay if necessary

}

};

</script>