

Import vuex modules from another file

New store-file.js:

```
const state = {}
const mutations = {}
const actions = {}
const getters = {
  get ter Name: (state) => {
    return state.a rray
  }
}
export default {
  nam esp aced: true
  state,
  mut ations,
  act ions,
  getters
}
```

index.js (main store file)

```
import store from './sto re- -
file'
const Store = new Vuex.S tore({
  mod ules: {
    store
  }
})
```

compon ent.vue:

```
<sc rip t>
  import { mapGetters } from
  'vuex'
  export default {
    com puted: {
      ...m ap Get ter s(' -
sto reN ame', ['gett erN ame'])
    }
  }
```

Import vuex modules from another file (cont)

```
> }
</script>
```

Getter to retrieve all array

```
const getters = {
  get ter Name: (state) => {
    return state.a rray
  }
}
```

Getter returning only certain elements

```
const getters = {
  tas ksTodo: state => {
    let tasks = {}
    Obj ect.ke ys( sta -
te.t as ks).fo rEa ch( fun ction
(key) {
      let task = state.t -
as ks[key]
      con sol e.l -
og( 'task: ', task)
      if (!task.co mpl -
eted) {
        tas ks[key] =
task
      }
    })
    return tasks;
  }
};
```

Deleting using vuex

component.vue where the method is triggered

```
<sc rip t>
  import { mapActions }
  from 'vuex'
  methods: {
    ...m ap Act ion s(' sto -
reN ame', ['acti onN ame'])
  }
```

Deleting using vuex (cont)

```
> promptToDelete() {
  this.$q.dialog({
    title: 'Confirm',
    message: 'Are You sure?',
    persistent: true }).onOk(() => {
    this.deleteFood(this.food.id)
  })
}
```

quasar.conf.js

```
plugins: [
  'Dialog'
]
```

store.js

```
import Vue from 'vue'
const actions = {
  deleteFood({ commit }, id) {
    commit('deleteFood', id)
  }
}
const mutations = {
  deleteFood(state, id) {
    let index = state.foods.findIndex(
      (food) => food.id == id)
    Vue.delete(state.foods, index)
  }
}
```



By przemekgrenda

cheatography.com/przemekgrenda/

Not published yet.

Last updated 6th January, 2020.

Page 1 of 2.

Sponsored by [Readable.com](https://readable.com)

Measure your website readability!

<https://readable.com>

Basic delete/add/edit operations

```
deleteFood: (state, id) => {
    let index = state.foods.findIndex(food => food.id === id);
    Vue.delete(state.foods, index);
},
addFood: (state, food) => {
    state.foods.push(food);
},
editFood: (state, food) => {
    let index = state.foods.findIndex(elem => elem.id === food.id);
    console.log(food, index);
    Vue.set(state.foods, index, food);
}
```

Quasar input validation

```
<q-input
  :rules="[
    val => val.length < 21 ||
    'Please use maximum of 20
    character s'
  ]"
  filled
  v-model="foodToSubmit.name"
  label="Name (e.g.
  Burger )"
  class="col" />
```

Validate requirements on submit

```
<q-input
  :rules="[
    val => val.length < 21 ||
    'Please use maximum of 20
    character s'
  ]"
  ref="name"
  />
<q-btn
  @click="submitForm"
  label="Save"
  color="primary"
  v-close-popup />
methods: {
  submitForm() {
    this.$refs.name.validate()

    if (!this.$refs.name.hasError) {
      this.submitFood()
    }
  },
  submitFood() {
    [...]
  }
}
```

Populating fields with data

```
mounted() {
  if (this.type === "edit") {
    console.log(this.food);
  }
}
```

Conditional displaying of tag

```
v-if="Object.keys(tasksTodo).length"
```

Computed with setter and getter

```
computed: {
  ...mapState("tasks", ["search"]),
  searchField: {
    get() {
    },
    set(value) {
    }
  }
}
```