

### Import vuex modules from another file

#### New store-file.js:

```
const state = {}
const mutations = {}
const actions = {}
const getters = {
  getName: (state) => {
    return state.array
  }
}
export default {
  namespaced: true
  state,
  mutations,
  actions,
  getters
}
```

#### index.js (main store file)

```
import store from './store-file'
const Store = new Vuex.Store({
  modules: {
    store
  }
})
```

#### component.vue:

```
<script>
  import { mapGetters } from
  'vuex'
  export default {
    computed: {
      ...mapGetters('storeName',
        ['getName'])
    }
  }
</script>
```

### Import vuex modules from another file (cont)

```

}
</script>
```

### Getter to retrieve all array

```
const getters = {
  getName: (state) => {
    return state.array
  }
}
```

### Getter returning only certain elements

```
const getters = {
  tasksTodo: state => {
    let tasks = {}
    Object.keys(state.tasks).fo-
    rEach(function (key) {
      let task = state.tas-
      ks[key]
      console.log('task: ',
        task)
      if (!task.completed) {
        tasks[key] = task
      }
    })
    return tasks;
  }
};
```

### Deleting using vuex

#### component.vue where the method is triggered

```
<script>
  import { mapActions } from
  'vuex'

  methods: {
    ...mapActions('storeName',
      ['actionName'])
  }
</script>
```

### Deleting using vuex (cont)

```

    promptToDelete() {
      this.$q.dialog({
        title:
'Confirm',
        message: 'Are
You sure?',
        persistent: true
      }).onOk(() => {
        this.dele-
        teFood(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.foo-
    ds.findIndex( (food) => food.id
    == id)
    Vue.delete(state.foods,
    index)
  }
}
```



By przemekgrenda

[cheatography.com/przemekgrenda/](https://cheatography.com/przemekgrenda/)

Not published yet.

Last updated 6th January, 2020.

Page 1 of 2.

Sponsored by [ApolloPad.com](https://apollopad.com)

Everyone has a novel in them. Finish Yours!

<https://apollopad.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 characters"
  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 characters"
  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) {
    }
  }
}
```