

# VUEX

---



The store for Vue.js.

It serves as a centralized store for all the components in an application

# USING VUEX #1

---

CDN:

```
<head>
<script
  src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"
</script>
<script
  src=" https://unpkg.com/vuex@3.6.2/dist/vuex.js "
</script>
</head>
```

# USING VUEX #2

---

CLI:

```
$ vue add vuex
```

main.js

```
import Vue from 'vue'  
import App from './App.vue'  
import store from './store'  
new Vue({store, render: h => h(App)}).$mount('#app')
```

# USING VUEX #3

---

## store/index.js

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

Vue.use(Vuex)

export default new Vuex.Store({
  state: {
    count: 0,
    orders: [],
  },
  getters: {
    getCount (state) {
      return state.count
    }
  },
  mutations: {
    increment (state, n) {
      state.count += n
    },
    addOrders(state, orders){
      state.orders = orders
    },
  },
},
```

# USING VUEX #4

---

```
actions: {
  async getMyOrdersFromDB({commit},id) {
    try {
      const response = await fetch(`/API/orders.php?id=${id}`)

      const data = await response.json()
      commit('addOrders', data)
    }
    catch (error) {
      console.log('error: ', error)
    }
  },
}
})
```

# USING VUEX IN A VIEW OR COMPONENT

---

```
export default {  
  computed: {  
    nextCount() {  
      return this.$store.getters.getCount;  
    }  
  },  
  methods: {  
    updateCount (n) {  
      this.$store.commit('increment', n)  
    },  
  },  
  created() {  
    this.$store.dispatch('getMyOrdersFromDB', id)  
  },  
}
```

# VUEX GETTER WITH PARAMETER #1

---

store/index.js

```
state: {
  products: []
},
getters: {
  getProducts (state) {
    return state.products
  },
  getProduct: (state) => (id) => {
    let index = state.products.findIndex(p => p.id == id)
    return state.products[index]
  },
},
```

# VUEX GETTER WITH PARAMETER #2

---

view or component

```
export default {
  data: function () {
    return {
      product: {}
    }
  },
  methods: {
    selectProduct() {
      this.product = this.$store.getters['products/getProduct'](this.$route.params.id)
    },
  }
}
```



# VUEX USING MULTIPLE STORES #1

---

store/index.js

```
import Vue from 'vue'  
import Vuex from 'vuex'  
import basket from './basket'  
import dishes from './dishes'  
import user from './user'  
import orders from './orders'  
  
Vue.use(Vuex)  
  
export default new Vuex.Store({  
  modules: {  
    basket,  
    dishes,  
    user,  
    orders  
  }  
})
```

# VUEX USING MULTIPLE STORES #2

---

## store/dishes

```
const dishes = {
  namespaced: true,
  state: {
    dishes: []
  },
  getters: {
    getDishes (state) {
      return state.dishes;
    },
  },
  mutations: {
    addDishes(state, dishes){
      state.dishes = dishes
    },
  },
  actions: {
  },
}
export default
dishes
```

# VUEX USING MULTIPLE STORES #3

---

view or component

```
export default {
  data() {
    return {
      dishes: [],
    }
  },
  mounted() {
    this.getDishes()
  },
  methods: {
    getDishes() {
      this.dishes = this.$store.getters['dishes/getDishes']
    },
    orderDish(id) {
      this.$store.commit('basket/incrementDish', id)
    },
  },
}
```