

Vue

- [【Vue】相關連結](#)
- [vue](#)
- [Vue 建立專案](#)

【Vue】相關連結

- [前言 | 重新認識 Vue.js | Kuro Hsu](#)
- [\[重構倒數30天，你的網站不Vue白不Vue \] :: 2021 iThome 鐵人賽](#)
- [Vue.js 進階心法 :: 2021 iThome 鐵人賽](#)
- [介紹 | Vue Router \(vuejs.org\)](#)

-
- [【UI】Why Quasar? | Quasar Framework](#)
 - [【UI】Element Plus \(element-plus.org\)](#)
 - [【UI】vuetifyjs](#)
-

vue

```
routes:[
  {
    path: '/login',
    name: 'login',
    component: loginPage,
    beforeEnter: (to, from, next) => {
      Store.dispatch('role/getMe').then(response => {
        next()
      })
    }
  },
]
```

=====

```
import Vue from 'vue'
import Vuex from 'vuex'
import sms from './modules/sms'
import role from './modules/role'
import group from './modules/group'
import permission from './modules/permission'
```

```
Vue.use(Vuex)
```

```
const store = new Vuex.Store({
  modules: {
    sms: sms,
    role: role,
    group: group,
    permission: permission
  }
})
```

```
export default store
```

=====

```
import axios from 'axios'
```

```
const state = {
  permissionList: [],
  permissionDropdownList: [],
  permissionDetail: {}
}
```

```
const actions = {
  async getList ({ commit }, params) {
    try {
      const { data: { data, total } } = await axios.get(`api/permissions`, { params })
      commit('SET_PERMISSION_LIST', {
        data,
        total: total
      })
    } catch (err) {
      console.log('get list', err)
    }
  }
}
```

```

    },

    async getDropDownList ({ commit }, params) {
      try {
        const { data: { data, total } } = await axios.get(`api/permissions/dropdownlist`, { params })
        commit('SET_PERMISSION_DROPDOWNLIST', {
          data,
          total: total
        })
      } catch (err) {
        console.log('get dropdown list', err)
      }
    },
  },

}

const mutations = {
  SET_PERMISSION_LIST (state, permissions) {
    state.permissionList = permissions.data
  },
  SET_PERMISSION_DROPDOWNLIST (state, permissions) {
    state.permissionDropDownList = permissions.data
  },
  SET_PERMISSION_DETAIL (state, permission) {
    state.permissionDetail = permission.data.data
  },
}

export default {
  namespaced: true,
  state,
  mutations,
  actions
}

```

Vue 建立專案

```
# install vue cli
npm install -g @vue/cli

# 建立 my-project 專案
vue create my-project
```

選第三個

```
Vue CLI v4.5.15
? Please pick a preset:
  Default ([Vue 2] babel, eslint)
  Default (Vue 3) ([Vue 3] babel, eslint)
> Manually select features
```

按下空白選取或是取消組件，Enter

```
Vue CLI v4.5.15
? Please pick a preset: Manually select features
? Check the features needed for your project:
  ● Choose Vue version
  ● Babel
  ○ TypeScript
  ○ Progressive Web App (PWA) Support
> ● Router
  ○ Vuex
  ● CSS Pre-processors
  ○ Linter / Formatter
  ○ Unit Testing
  ○ E2E Testing
```

選擇vue版本 -> 3.X

```
Vue CLI v4.5.15
? Please pick a preset: Manually select features
? Check the features needed for your project: Choose Vue version, Babel, Router, CSS Pre-processors
? Choose a version of Vue.js that you want to start the project with
  2.x
> 3.x
```

Yes -> Sass/SCSS(with node-sass) -> Inpackage.json -> Yes -> Enter

```
Vue CLI v4.5.15
? Please pick a preset: Manually select features
? Check the features needed for your project: Choose Vue version, Babel, Router, CSS Pre-processors
? Choose a version of Vue.js that you want to start the project with 3.x
? Use history mode for router? (Requires proper server setup for index fallback in production) Yes
? Pick a CSS pre-processor (PostCSS, Autoprefixer and CSS Modules are supported by default): Sass/SCSS (with node-sass)
? Where do you prefer placing config for Babel, ESLint, etc.? In package.json
? Save this as a preset for future projects? Yes
? Save preset as: 
```

進入目錄 -> 起測試服務

```
# 進入專案目錄
cd vue-demo
```

```
# 起測試服務  
npm run serve
```



Successfully created project **vue-demo**.
Get started with the following commands:

```
$ cd vue-demo  
$ npm run serve
```