

16-1 引入封装的请求库和 Vuex

[目录结构](#)

[封装 request.js](#)

[使用 Vuex](#)

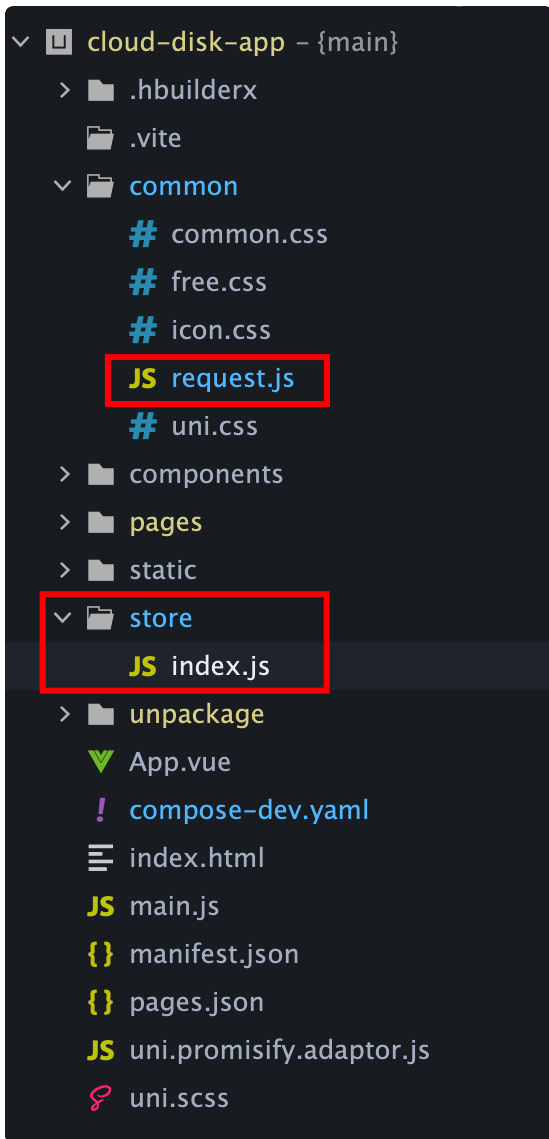
[全局引入 Vuex](#)

[测试](#)

请求库的封装，认真看一下封装的细节

目录结构

- 如图新建相关目录和文件



封装 request.js

- 全局定义了请求的 API 的基地址、请求头类型、请求数据、默认 GET 请求、默认返回 JSON 数据, token 等;
- request 方法封装了 promise, 将以上内容封装为入参 options 对象, 并统一做了有无 token 的处理, 和后端返回的数据类型完全对应, 并做好数据解析;
- 细化几种不同类型的请求, 方便调用;
- 封装了文件上传方法 upload。

```
1 export default {
2   // 全局配置
3   common: {
4     // #ifndef H5
5     baseUrl: "http://127.0.0.1:7001",
6     // #endif
7     // #ifdef H5
8     baseUrl: "/api",
9     // #endif
10    header: {
11      'Content-Type': 'application/json;charset=UTF-8',
12    },
13    data: {},
14    method: 'GET',
15    dataType: 'json',
16    token: false
17  },
18  // 请求返回promise
19  request(options = {}) {
20    // 组织参数
21    options.url = this.common.baseUrl + options.url
22    options.header = options.header || this.common.header
23    options.data = options.data || this.common.data
24    options.method = options.method || this.common.method
25    options.dataType = options.dataType || this.common.dataType
26    options.token = options.token === true ? true : this.common.token
27    // 请求
28    return new Promise((res, rej) => {
29      // 请求之前验证...
30      // token验证
31      if (options.token) {
32        let token = uni.getStorageSync('token')
33        // 二次验证
34        if (!token && options.noJump !== true) {
35          uni.showToast({
36            title: '请先登录',
37            icon: 'none'
38          });
39          // token不存在时跳转
40          uni.navigateTo({
41            url: '/pages/login/login',
42          });
43          return rej("请先登录")
44        }
45        // 往header头中添加token
```

```

46         options.header.token = token
47     }
48
49     // 请求中...
50     uni.request({
51         ...options,
52         success: (result) => {
53             // 返回原始数据
54             if (options.native) {
55                 return res(result)
56             }
57             // 服务端失败
58             if (result.statusCode !== 200) {
59                 if (options.toast !== false) {
60                     uni.showToast({
61                         title: result.data.data || '服务端失败',
62                         icon: 'none'
63                     });
64                 }
65                 // token不合法, 直接退出登录
66                 if (result.data.data === 'Token 令牌不合法!') {
67                     // 退出登录操作
68                     // $store.dispatch('logout')
69                 }
70                 return rej(result.data)
71             }
72             // 其他验证...
73             // 成功
74             let data = result.data.data
75             res(data)
76         },
77         fail: (error) => {
78             uni.showToast({
79                 title: error.errMsg || '请求失败',
80                 icon: 'none'
81             });
82             return rej(error)
83         }
84     });
85 }
86 },
87 // get请求
88 get(url, options = {}) {
89     options.url = url
90     options.data = {}
91     options.method = 'GET'
92     return this.request(options)
93 },

```

```

94 // post请求
95 post(url, data = {}, options = {}) {
96   options.url = url
97   options.data = data
98   options.method = 'POST'
99   return this.request(options)
100 },
101 // delete请求
102 del(url, data = {}, options = {}) {
103   options.url = url
104   options.data = data
105   options.method = 'DELETE'
106   return this.request(options)
107 },
108 // 上传文件
109 upload(url, data, onProgress = false) {
110   return new Promise((result, reject) => {
111     // 上传
112     let token = uni.getStorageSync('token')
113     if (!token) {
114       uni.showToast({
115         title: '请先登录',
116         icon: 'none'
117       });
118       // token不存在时跳转
119       return uni.reLaunch({
120         url: '/pages/login/login',
121       });
122     }
123
124     const uploadTask = uni.uploadFile({
125       url: this.common.baseUrl + url,
126       filePath: data.filePath,
127       name: data.name || "files",
128       header: {
129         token
130       },
131       formData: data.formData || {},
132       success: (res) => {
133         if (res.statusCode !== 200) {
134           result(false)
135           return uni.showToast({
136             title: '上传失败',
137             icon: 'none'
138           });
139         }
140         let message = JSON.parse(res.data)
141         result(message.data);

```

```

142     },
143     fail: (err) => {
144         console.log(err);
145         reject(err)
146     }
147 })
148     //回调，更新上传进度
149     uploadTask.onProgressUpdate((res) => {
150         if (typeof onProgress === 'function') {
151             onProgress(res.progress)
152         }
153     });
154 })
155 }
156 }

```

使用 Vuex

store/index.js , 引入 Vuex 状态保持。

<https://vuex.vuejs.org/zh/>

```

1  import { createStore } from 'vuex'
2
3  const store = createStore({
4    state: {
5      username: 'zhangsan',
6    },
7  })
8
9  export default store

```

全局引入 Vuex

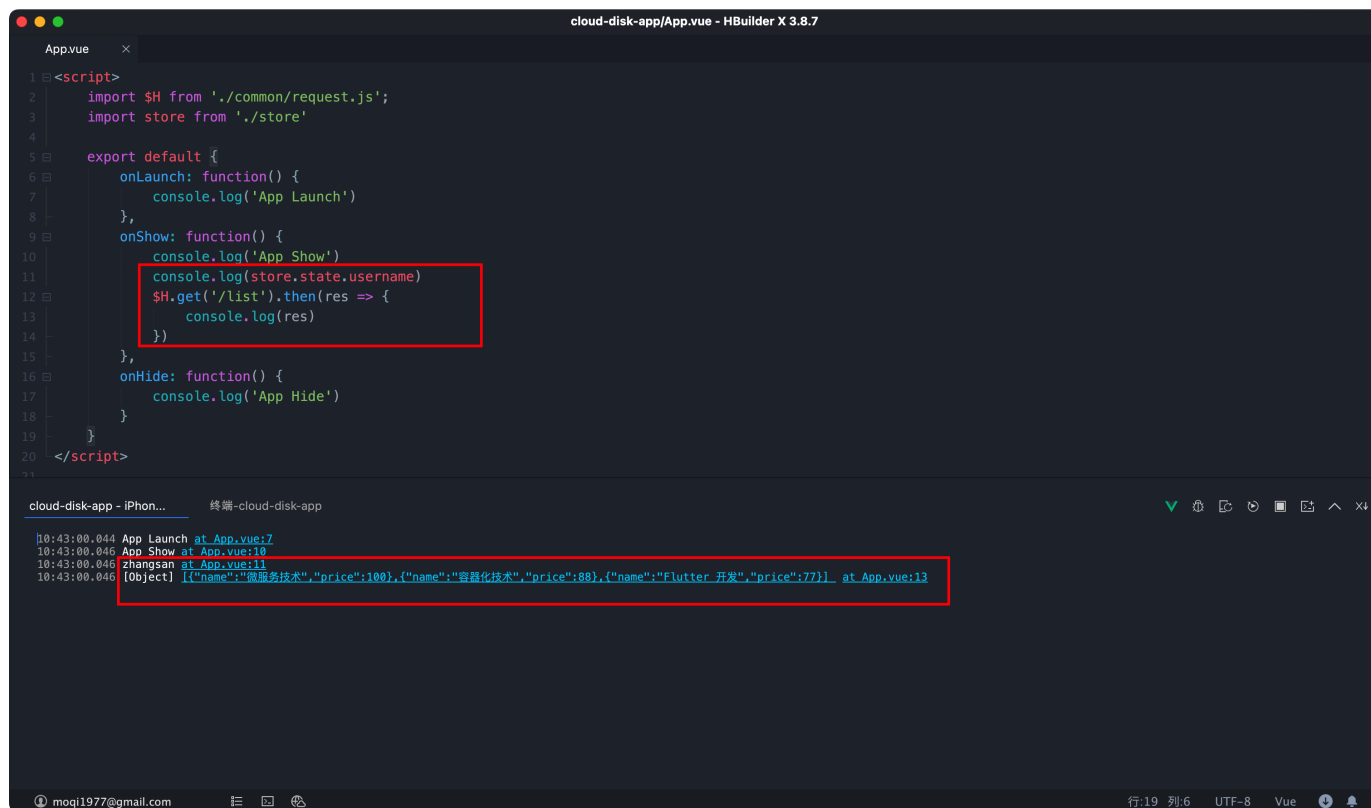
在 main.js 中全局引入 Vuex

main.js

```
1 import App from './App'
2 import { createSSRApp } from 'vue'
3
4 // 引入 Vuex
5 import store from './store'
6
7 export function createApp() {
8   const app = createSSRApp(App)
9
10  app.use(store)
11  return {
12    app
13  }
14 }
```

测试

App.vue 在应用启动的时候打印一下之前存入 store 的状态的值，并请求后端的 `http://127.0.0.1:7001/list` 接口，观察控制台输出结果。



The screenshot shows the HBuilder X IDE with the 'App.vue' file open. The code in the script tag is as follows:

```
1 <script>
2   import $H from './common/request.js';
3   import store from './store'
4
5   export default {
6     onLaunch: function() {
7       console.log('App Launch')
8     },
9     onShow: function() {
10      console.log('App Show')
11      console.log(store.state.username)
12      $H.get('/list').then(res => {
13        console.log(res)
14      })
15    },
16    onHide: function() {
17      console.log('App Hide')
18    }
19  }
20 </script>
```

The console output at the bottom shows the following logs:

```
10:43:00.044 App Launch at App.vue:7
10:43:00.046 App Show at App.vue:10
10:43:00.046 zhangsan at App.vue:11
10:43:00.046 [Object] [{"name":"微服务技术","price":100}, {"name":"容器化技术","price":88}, {"name":"Flutter 开发","price":77}] at App.vue:13
```

```
commit fb080af8f436fe97f2e74966ae54d041e052df7c (HEAD -> main)
Author: mqxu <moqi1977@gmail.com>
Date: Sat Jul 8 20:52:57 2023 +0800
```

引入封装的请求库和 Vuex