



# 插件篇

加微信：[talkingcoder](#) 进群讨论，注明“插件”

# 关于我



Aresn

-  TalkingData

-  TalkingCoder

-  iView

-  《Vue.js 实战》



- Vue.js 实战之 Render 函数 ✓
- Vue.js 实战之组件篇(基础) ✓
- Vue.js 实战之工程篇 ✓
- Vue.js 实战之插件篇 (vuex、vue-router、axios)
- iView 实战教程系列

- 新书预告

《Vue.js实战》将在9月由清华大学出版社出版，敬请期待！

- Vue.js 插件

注册插件需要一个公开的方法 `install`，它的第一个参数是 `Vue` 构造器，第二个参数是一个可选的选项对象。

```
1  Vue.use(MyPlugin)
2  // 或
3  Vue.use(MyPlugin, {
4    // 参数
5  })
```

```
1  MyPlugin.install = function (Vue, options) {
2    // 全局注册组件 (指令等功能资源类似)
3    Vue.component('component-name', {
4      // 组件内容
5    })
6    // 添加实例方法
7    Vue.prototype.$Notice = function () {
8      // 逻辑...
9    }
10   // 添加全局方法或属性
11   Vue.globalMethod = function () {
12     // 逻辑...
13   }
14   // 添加全局混合
15   Vue.mixin({
16     mounted: function () {
17       // 逻辑...
18     }
19   })
20 }
```

- 前端路由与 vue-router

**什么是前端路由？**

- vue-router 基本用法

***Code***

- vue-router 跳转

- <router-link>

- 程式导航 (push、replace、go)

- vue-router 高级用法

- 设置标题
- 返回顶端
- 全局 Loading

- Vuex 状态管理与使用场景

数据 (model) 和视图 (view)

```
1 <template>
2   <div>
3     {{ message }}
4     <button @click="handleClick">Change word</button>
5   </div>
6 </template>
7 <script>
8   export default {
9     data () {
10      return {
11        message: 'Hello World.'
12      };
13    },
14    methods: {
15      handleClick () {
16        this.message = 'Hello Vue.';
17      }
18    }
19  };
20 </script>
```

- Vuex 基本用法

***Code***

- Vuex 高级用法

***Code***

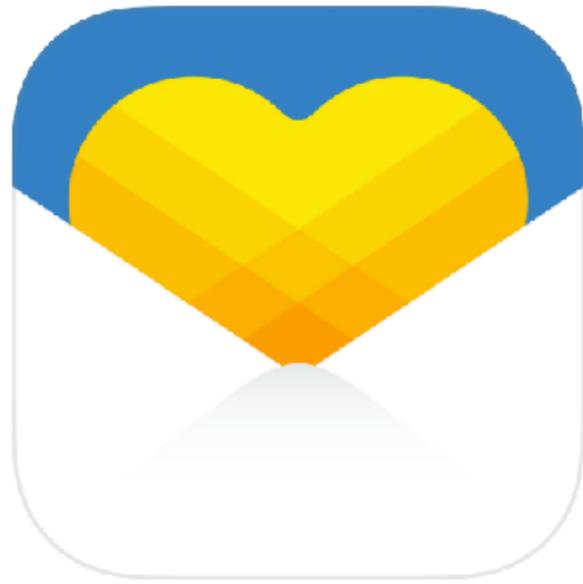
- 实战：中央事件总线插件 vue-bus

***Code***

## ● 练习

学习 XMLHttpRequest (即 xhr)  
相关知识, 开发一个简单的 \$ajax  
插件, 用于异步获取服务端数据。  
以下实现 ajax 的代码可以作为参  
考:

```
1  const ajax = function (options = {}) {
2      options.type = (options.type || 'GET').toUpperCase();
3
4      let data = [];
5      for(let i in options.data){
6          data.push(encodeURIComponent(i) +
7              '=' + encodeURIComponent(options.data[i]));
8      }
9      data = data.join('&');
10
11     const xhr = new XMLHttpRequest();
12     xhr.onreadystatechange = function () {
13         if (xhr.readyState === 4) {
14             const status = xhr.status;
15             if (status >= 200 && status < 300) {
16                 options.success &&
17                 options.success(JSON.parse(xhr.responseText), xhr.responseXML);
18             } else {
19                 options.error && options.error(status);
20             }
21         }
22     };
23     if (options.type === 'GET') {
24         xhr.open('GET', options.url + '?' + data, true);
25         xhr.send(null);
26     } else if (options.type === 'POST') {
27         xhr.open('POST', options.url, true);
28         xhr.setRequestHeader(
29             'Content-Type',
30             'application/x-www-form-urlencoded');
31         xhr.send(data);
32     }
33 };
```



Thanks