## 网页代码

1. <!DOCTYPE html>
2. <html lang='en'>
3. <head>
4. <meta charset='UTF-8'>
5. <meta name='viewport' content='width=device-width, initial-scale=1.0'>
6. <meta http-equiv='X-UA-Compatible' content='ie=edge'>
7. <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
8. <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@3.3.7/dist/css/bootstrap.min.css">
9. <title>各厂商手机价格展示</title>
10. </head>
11. <body>
12. <div id='app'>
13. <div class="container">
14. <h1>各厂商手机价格展示</h1>
15. <form class="form-inline">
16. <div class="form-group">
17. <label for="name">厂商：</label>
18. <input type="text" class="form-control" id="name" placeholder="请输入厂商" v-model="add\_book.name">
19. </div>
20. <div class="form-group">
21. <label for="author">型号：</label>
22. <input type="text" class="form-control" id="author" placeholder="请输入型号" v-model="add\_book.author">
23. </div>
24. <div class="form-group">
25. <label for="price">价格（单位：元）：</label>
26. <input type="number" class="form-control" id="price" placeholder="请输入价格" v-model="add\_book.price">
27. </div>
28. <button type="submit" class="btn btn-primary" @click.prevent="add">添加</button>
29. </form>
30. <br>
31. <form class="form-inline">
32. <div class="form-group">
33. <label for="keywords">搜索（厂商）关键字：</label>
34. <input type="text" class="form-control" id="keywords" placeholder="请输入关键字" v-model="keywords">
35. </div>
36. </form>
37. <table class="table">
38. <thead>
39. <tr>
40. <th>序号</th>
41. <th>厂商</th>
42. <th>型号</th>
43. <th>价格（单位：元）</th>
44. <th>插入时间</th>
45. <th>操作</th>
46. </tr>
47. </thead>
48. <tbody>
49. <tr v-for="(book,index) in books\_result">
50. <td>{{index+1}}</td>
51. <td>{{book.name}}</td>
52. <td>{{book.author}}</td>
53. <td>{{book.price}}</td>
54. <td>{{book.atime|timeFormat}}</td>
55. <td><button class="btn btn-danger" @click="del(index)">删除</button></td>
56. </tr>
57. </tbody>
58. </table>
59. </div>
60. </div>
61. <script>
62. new Vue({
63. el: '#app',
64. data: {
65. books: [
66. { "name": "OPPO", "author": "OPPO Find X6 Pro (16GB + 512GB)", "price": 6999, "atime": new Date() },
67. { "name": "Vivo", "author": "Vivo X90 Pro (16GB + 512GB)", "price": 5999 , "atime": new Date() },
68. { "name": "OnePlus", "author": "OnePlus 7 Pro (12GB + 256GB)", "price": 4999 , "atime": new Date() },
69. { "name": "Xiaomi", "author": "MIX (4GB + 128GB)", "price": 2900 , "atime": new Date() },
70. { "name": "Apple", "author": "IPhone X (3GB + 256GB)", "price": 6888 , "atime": new Date() },
71. { "name": "Xiaomi", "author": "Xiaomi 13 Ultra (16GB + 1024GB)", "price": 7299 , "atime": new Date() },
72. { "name": "Xiaomi", "author": "Redmi K20 Pro (12GB + 512GB)", "price": 3199 , "atime": new Date() },
73. { "name": "OPPO", "author": "OPPO Reno Ace (12GB + 256GB)", "price": 3799 , "atime": new Date() },
74. ],
75. add\_book: {
76. name: '',
77. author: '',
78. price: 0,
79. },
80. keywords: "",
81. },
83. computed: {
84. books\_result() {
85. var \_this = this;
86. if (this.keywords) {

89. let newBooks = this.books.filter(function (item) {
90. if (item.name.indexOf(\_this.keywords) >= 0) {
92. return true
93. } else {
94. return false
95. }
96. })
97. return newBooks
98. } else {
99. return this.books //没有关键词就显示所有的数据
100. }
101. }
102. },
103. methods: {
104. // 添加型号记录
105. add() {
106. // let book = this.add\_book
108. let book = JSON.parse(JSON.stringify(this.add\_book));
109. book.atime = new Date();//获取当前时间
110. this.books.push(book);
111. },
112. // 删除指定的型号记录
113. del(index) {
114. this.books.splice(index, 1)
115. }
116. },
118. filters: {
119. timeFormat: function (value) {
120. value = new Date(value);
121. let year = value.getFullYear();
122. let month = value.getMonth() + 1;
123. let day = value.getDate();
124. let hour = value.getHours();
125. let minutes = value.getMinutes();
126. return `${year}-${month}-${day} ${hour}:${minutes}`
127. }
128. }
129. })
130. </script>
131. </body>
132. </html>

## 网页说明

### 方法1：添加手机型号信息

<methods: {

        // 添加型号记录

        add() {

          // let book = this.add\_book

          //JSON.stringify() 方法用于将js对象转换为 JSON 字符串，JSON.parse()方法用于将一个 JSON 字符串转换为js对象。

          let book = JSON.parse(JSON.stringify(this.add\_book));

          book.atime = new Date();//获取当前时间

          this.books.push(book);

        },

        // 删除指定的型号记录

        del(index) {

          this.books.splice(index, 1)

        }

      },

Button中写一个事件，事件名叫abb，abb在methods方法中定义，用push()方法添加型号

### 方法2：删除手机信息

 // 删除指定的型号记录

        del(index) {

          this.books.splice(index, 1)

        }

      },

Button中写一个事件，事件名叫del，del在methods方法中定义，用splice()方法删除当前数据index,index指代内容1是个数

### 方法3：关键字搜索

      computed: {

        books\_result() {

          var \_this = this;

          if (this.keywords) {

         //this的作用域到此结束，无法穿透进入filter函数,所以需要用一个变量（比如\_this）来保存this的指向。

            // 下面的参数item指的是其中一条图书的搜索记录

            let newBooks = this.books.filter(function (item) {

              if (item.name.indexOf(\_this.keywords) >= 0) {

                //找到输入的关键词对应的书名的索引，存在则>=0,不存在则返回-1

                return true

              } else {

                return false

              }

            })

            return newBooks

          } else {

            return this.books //没有关键词就显示所有的数据

          }

        }

      },

Input中写一个事件，事件名叫keywords，keywords在computed方法中定义，在keywords定义一个item搜索记录，找到对应的车辆索引，存在>=0,不存在返回-1，没有关键字就显示所有数据

## 网页截图

### 初始界面



### 添加一条数据界面



### 删除数据界面



### 关键字搜索

