

一、分页列表

1、定义api

创建文件 src/api/yygh/hospset.js

```
1 import request from '@/utils/request'
2
3 const api_name = '/admin/hosp/hospitalSet'
4 export default {
5   //医院设置列表
6   getPageList(page, limit, searchObj) {
7     return request({
8       url: `${api_name}/findPageCondition/${page}/${limit}`,
9       method: 'post',
10      data: searchObj
11    })
12  }
13 }
```

2、初始化vue组件

src/views/yygh/hospset/list.vue

```
1 <template>
2   <div class="app-container">
3     医院设置列表
4   </div>
5 </template>
6
7 <script>
8 import hospset from '@api/yygh/hospset'
9 export default {
10
11   data() { // 定义数据
12     return {}
13   },
```

```

14
15 created() { // 当页面加载时获取数据
16     this.fetchData()
17 },
18
19 methods: {
20     fetchData() { // 调用api层获取数据库中的数据
21         console.log('加载列表')
22     }
23 }
24 }
25 </script>

```

3、定义data

```

1 data() {
2     return {
3         listLoading: true, // 是否显示loading信息
4         list: null, // 数据列表
5         total: 0, // 总记录数
6         page: 1, // 页码
7         limit: 10, // 每页记录数
8         searchObj: {} // 查询条件
9     }
10 },

```

4、定义methods

```

1 methods: {
2     fetchData(page = 1) { // 调用api层获取数据库中的数据
3         console.log('加载列表')
4         this.page = page
5         this.listLoading = true
6         hospset.getPageList(this.page, this.limit, this.searchObj).then(response => {
7             // debugger 设置断点调试
8             if (response.success === true) {
9                 this.list = response.data.rows

```

```

10         this.total = response.data.total
11     }
12     this.listLoading = false
13 })
14 },
15 }

```

5、表格渲染

```

1 <el-table
2     v-loading="listLoading"
3     :data="list"
4     element-loading-text="数据加载中"
5     border
6     fit
7     highlight-current-row>
8
9     <el-table-column
10         label="序号"
11         width="70"
12         align="center">
13         <template slot-scope="scope">
14             {{ (page - 1) * limit + scope.$index + 1 }}
15         </template>
16     </el-table-column>
17
18     <el-table-column prop="hosname" label="医院名称" width="180" />
19
20     <el-table-column prop="hoscode" label="医院编号" width="160" />
21
22     <el-table-column prop="apiUrl" label="地址" width="200"/>
23
24     <el-table-column prop="contactsName" label="联系人"/>
25
26     <el-table-column prop="status" label="状态">
27         <template slot-scope="scope">
28             {{ scope.row.status===1?'可用':'不可用' }}
29         </template>
30     </el-table-column>
31

```

```

32     <el-table-column label="操作" width="200" align="center">
33       <template slot-scope="scope">
34         <router-link :to="'/yygh/hospset/edit/'+scope.row.id">
35           <el-button type="primary" size="mini" icon="el-icon-edit">修改</el-bu
36         </router-link>
37         <el-button type="danger" size="mini" icon="el-icon-delete" @click="remove
38       </template>
39     </el-table-column>
40 </el-table>

```

6、分页组件

```

1 <!-- 分页 -->
2 <el-pagination
3     :current-page="page"
4     :page-size="limit"
5     :total="total"
6     style="padding: 30px 0; text-align: center;"
7     layout="total, prev, pager, next, jumper"
8     @current-change="fetchData"
9     />

```

7、顶部查询表单

```

1 <!-- 查询表单 -->
2 <el-form :inline="true" class="demo-form-inline">
3   <el-form-item>
4     <el-input v-model="searchObj.hosname" placeholder="医院名称"/>
5   </el-form-item>
6
7   <el-form-item>
8     <el-input v-model="searchObj.hoscode" placeholder="医院编号"/>
9   </el-form-item>
10
11   <el-button type="primary" icon="el-icon-search" @click="fetchData()">查询</el-but
12   <el-button type="default" @click="resetData()">清空</el-button>
13 </el-form>

```

清空方法

```
1 resetData() {  
2     this.searchObj = {}  
3     this.fetchData()  
4 }
```

8、测试

二、删除

1、定义api

src/api/yygh/hospset.js

```
1 removeById(id) {  
2     return request({  
3         url: `${api_name}/${id}`,  
4         method: 'delete'  
5     })  
6 }
```

2、定义methods

src/views/yygh/hospset/list.vue

使用MessageBox 弹框组件

```
1 removeDataById(id) {  
2     // debugger  
3     // console.log(memberId)  
4     this.$confirm('此操作将永久删除该记录，是否继续?', '提示', {  
5         confirmButtonText: '确定',  
6         cancelButtonText: '取消',
```

```
7       type: 'warning'
8   }).then(() => {
9       return hospset.removeById(id)
10  }).then(() => {
11      this.fetchData()
12      this.$message({
13          type: 'success',
14          message: '删除成功!'
15      })
16  }).catch((response) => { // 失败
17      if (response === 'cancel') {
18          this.$message({
19              type: 'info',
20              message: '已取消删除'
21          })
22      } else {
23          this.$message({
24              type: 'error',
25              message: '删除失败'
26          })
27      }
28  })
29 }
```