

一、首页数据显示

1、医院列表显示（接口）

1.1 service接口与实现

在管理平台 医院分页列表时已经提供，目前我们可以直接使用

1.2 添加controller方法

创建HospitalApiController

```
1 @Api(tags = "医院显示接口")
2 @RestController
3 @RequestMapping("/api/hosp/hospital")
4 public class HospitalApiController {
5
6     @Autowired
7     private HospitalService hospitalService;
8
9     @ApiOperation(value = "获取分页列表")
10    @GetMapping("/{page}/{limit}")
11    public R index(
12        @PathVariable Integer page,
13        @PathVariable Integer limit,
14        HospitalQueryVo hospitalQueryVo) {
15
16        Page<Hospital> pageModel = hospitalService.selectHospPage(page, limit, hospit
17        return R.ok().data("pages",pageModel);
18    }
19 }
```

2、医院搜索（接口）

2.1 service接口与实现

在HospitalService添加接口和实现

```
1  /**
2      * 根据医院名称获取医院列表
3      */
4  List<Hospital> findByHosname(String hosname);
5
6  @Override
7  public List<Hospital> findByHosname(String hosname) {
8      return hospitalRepository.findHospitalByHosnameLike(hosname);
9  }
```

在HospitalRepository类添加方法

```
1  List<Hospital> findHospitalByHosnameLike(String hosname);
```

2.2 .HospitalApiController 添加方法

```
1  @ApiOperation(value = "根据医院名称获取医院列表")
2  @GetMapping("findByHosname/{hosname}")
3  public R findByHosname(
4      @ApiParam(name = "hosname", value = "医院名称", required = true)
5      @PathVariable String hosname) {
6
7      List<Hospital> list = hospitalService.findByHosname(hosname);
8      return R.ok().data("list", list);
9  }
```

3、首页数据前端

3.1 封装api

(1) 创建api文件夹，创建/api/yygh/hospital.js

```
1 import request from '@utils/request'
2
3 const api_name = `/api/hosp/hospital`
4
5 export default {
6   getPageList(page, limit, searchObj) {
7     return request({
8       url: `${api_name}/${page}/${limit}`,
9       method: 'get',
10      params: searchObj
11    })
12  },
13  getByHosname(hosname) {
14    return request({
15      url: `${api_name}/findByHosname/${hosname}`,
16      method: 'get'
17    })
18  }
19 }
```

(2) 创建/api/yygh/dict.js

```
1 import request from '@utils/request'
2
3 const api_name = '/admin/cmn/dict'
4
5 export default {
6   findByDictCode(dictCode) {
7     return request({
8       url: `${api_name}/findByDictCode/${dictCode}`,
9       method: 'get'
10    })
11  },
12   findByParentId(parentId) {
13     return request({
14       url: `${api_name}/findChildData/${parentId}`,
15       method: 'get'
16     })
17   }
18 }
```

3.2 添加组件

(1) 修改pages/index.vue组件

```

1 <template>
2   <div class="home page-component">
3     <el-carousel indicator-position="outside">
4       <el-carousel-item v-for="item in 2" :key="item">
5         
6       </el-carousel-item>
7     </el-carousel>
8     <!-- 搜索 -->
9     <div class="search-container">
10      <div class="search-wrapper">
11        <div class="hospital-search">
12          <el-autocomplete
13            class="search-input"
14            prefix-icon="el-icon-search"
15            v-model="hosname"
16            :fetch-suggestions="querySearchAsync"
17            placeholder="点击输入医院名称"
18            @select="handleSelect"
19          >
20            <span slot="suffix" class="search-btn v-link highlight clickable selected">搜
21          </el-autocomplete>
22        </div>
23      </div>
24    </div>
25    <!-- bottom -->
26    <div class="bottom">
27      <div class="left">
28        <div class="home-filter-wrapper">
29          <div class="title"> 医院</div>
30        <div>
31          <div class="filter-wrapper">
32            <span
33              class="label">等级: </span>
34            <div class="condition-wrapper">
35              <span class="item v-link clickable"

```

```

36         :class="hostypeActiveIndex == index ? 'selected' : ''"
37         v-for="(item,index) in hostypeList" :key="item.id"
38         @click="hostypeSelect(item.value, index)">{{ item.name }}</span>
39     </div>
40 </div>
41 <div class="filter-wrapper">
42     <span
43     class="label">地区: </span>
44     <div class="condition-wrapper">
45         <span class="item v-link clickable"
46             :class="provinceActiveIndex == index ? 'selected' : ''"
47             v-for="(item,index) in districtList" :key="item.id"
48             @click="districtSelect(item.value, index)">{{ item.name }}</span>
49     </div>
50 </div>
51 </div>
52 </div>
53 <div class="v-scroll-list hospital-list">
54     <div class="v-card clickable list-item" v-for="item in list" :key="item.id">
55         <div class="">
56             <div class="hospital-list-item hos-item" index="0" @click="show(item.hoscoo
57                 <div class="wrapper">
58                     <div class="hospital-tittle"> {{ item.hosname }}</div>
59                     <div class="bottom-container">
60                         <div class="icon-wrapper">
61                             <span class="iconfont">□</span>{{ item.param.hostypeString }}
62                         </div>
63                         <div class="icon-wrapper">
64                             <span class="iconfont">□</span>每天{{ item.bookingRule.releaseTime }}放号
65                         </div>
66                     </div>
67                 </div>
68                 
71             </div>
72         </div>
73     </div>
74 </div>
75 </div>
76 <div class="right">
77     <div class="common-dept">
78     <div class="header-wrapper">

```

```
79 <div class="title"> 常见科室</div>
80 <div class="all-wrapper"><span>全部</span>
81 <span class="iconfont icon">□</span>
82 </div>
83 </div>
84 <div class="content-wrapper">
85 <span class="item v-link clickable dark">神经内科 </span>
86 <span class="item v-link clickable dark">消化内科 </span>
87 <span class="item v-link clickable dark">呼吸内科 </span>
88 <span class="item v-link clickable dark">内科 </span>
89 <span class="item v-link clickable dark">神经外科 </span>
90 <span class="item v-link clickable dark">妇科 </span>
91 <span class="item v-link clickable dark"> 产科 </span>
92 <span class="item v-link clickable dark">儿科 </span>
93 </div>
94 </div>
95 <div class="space">
96 <div class="header-wrapper">
97 <div class="title-wrapper">
98 <div class="icon-wrapper"><span
99 class="iconfont title-icon">□</span>
100 </div>
101 <span class="title">平台公告</span>
102 </div>
103 <div class="all-wrapper">
104 <span>全部</span>
105 <span class="iconfont icon">□</span>
106 </div>
107 </div>
108 <div class="content-wrapper">
109 <div class="notice-wrapper">
110 <div class="point"></div>
111 <span class="notice v-link clickable dark">关于延长北京大学国际医院放假的通知 </span>
112 </div>
113 <div class="notice-wrapper">
114 <div class="point"></div>
115 <span class="notice v-link clickable dark">北京中医药大学东方医院部分科室医生门诊
116 </div>
117 <div class="notice-wrapper">
118 <div class="point"></div>
119 <span class="notice v-link clickable dark"> 武警总医院号源暂停更新通知 </span>
120 </div>
121 </div>
```

```

122     </div>
123     <div class="suspend-notice-list space">
124     <div class="header-wrapper">
125     <div class="title-wrapper">
126         <div class="icon-wrapper">
127             <span class="iconfont title-icon">□</span>
128         </div>
129         <span class="title">停诊公告</span>
130     </div>
131     <div class="all-wrapper">
132         <span>全部</span>
133         <span class="iconfont icon">□</span>
134     </div>
135 </div>
136 <div class="content-wrapper">
137     <div class="notice-wrapper">
138     <div class="point"></div>
139     <span class="notice v-link clickable dark"> 中国人民解放军总医院第六医学中心(原海5
140     </div>
141     <div class="notice-wrapper">
142     <div class="point"></div>
143     <span class="notice v-link clickable dark"> 首都医科大学附属北京潞河医院老年医学科门
144     </div>
145     <div class="notice-wrapper">
146         <div class="point"></div>
147         <span class="notice v-link clickable dark">中日友好医院中西医结合心内科门诊停诊
148     </div>
149     </div>
150 </div>
151 </div>
152 </div>
153 </div>
154 </template>
155 <script>
156
157 import hospApi from '@api/hosp/hospital.js'
158 import dictApi from '@api/hosp/dict.js'
159
160 export default {
161     //服务端渲染异步，显示医院列表
162     asyncData({ params, error }) {
163         //调用
164         return hospApi.getPageList(1,10,null)

```

```
165     .then(response => {
166         return {
167             list: response.data.pages.content,
168             pages: response.data.pages.totalPages
169         }
170     })
171 },
172 data() {
173     return {
174         searchObj: {},
175         page: 1,
176         limit: 10,
177
178         hosname: '', //医院名称
179         hostypeList: [], //医院等级集合
180         districtList: [], //地区集合
181
182         hostypeActiveIndex: 0,
183         provinceActiveIndex: 0
184     }
185 },
186 created() {
187     this.init()
188 },
189 methods:{
190     //查询医院等级列表 和 所有地区列表
191     init() {
192         //查询医院等级列表
193         dictApi.findByDictCode('Hostype')
194             .then(response => {
195                 //hostypeList清空
196                 this.hostypeList = []
197                 //向hostypeList添加全部值
198                 this.hostypeList.push({"name":"全部","value":""})
199                 //把接口返回数据，添加到hostypeList
200                 for(var i=0;i<response.data.list.length;i++) {
201                     this.hostypeList.push(response.data.list[i])
202                 }
203             })
204
205         //查询地区数据
206         dictApi.findByDictCode('Beijin')
207             .then(response => {
```



```
208         this.districtList = []
209         this.districtList.push({"name": "全部", "value": ""})
210         for(let i in response.data.list) {
211             this.districtList.push(response.data.list[i])
212         }
213     })
214 },
215
216 //查询医院列表
217 getList() {
218     hospApi.getPageList(this.page, this.limit, this.searchObj)
219     .then(response => {
220         for(let i in response.data.pages.content) {
221             this.list.push(response.data.pages.content[i])
222         }
223         this.page = response.data.pages.totalPages
224     })
225 },
226
227 //根据医院等级查询
228 hostypeSelect(hostype, index) {
229     //准备数据
230     this.list = []
231     this.page = 1
232     this.hostypeActiveIndex = index
233     this.searchObj.hostype = hostype
234     //调用查询医院列表方法
235     this.getList()
236 },
237
238 //根据地区查询医院
239 districtSelect(districtCode, index) {
240     this.list = []
241     this.page = 1
242     this.provinceActiveIndex = index
243     this.searchObj.districtCode = districtCode
244     this.getList();
245 },
246
247 //在输入框输入值，弹出下拉框，显示相关内容
248 querySearchAsync(queryString, cb) {
249     this.searchObj = []
250     if(queryString == '') return
```

```

251     hospApi.getByHosname(queryString).then(response => {
252         for (let i = 0, len = response.data.list.length; i < len; i++) {
253             response.data.list[i].value = response.data.list[i].hosname
254         }
255         cb(response.data.list)
256     })
257 },
258
259 //在下拉框选择某一个内容, 执行下面方法, 跳转到详情页面中
260 handleSelect(item) {
261     window.location.href = '/hospital/' + item.hoscode
262 },
263
264 //点击某个医院名称, 跳转到详情页面中
265 show(hoscode) {
266     window.location.href = '/hospital/' + hoscode
267 }
268 }
269 }
270 </script>

```

3.3 修改utils里面request.js路径

修改utils/request.js文件

```

1 // 请求地址端口号修改为 gateway网关地址
2 const service = axios.create({
3     baseURL: 'http://localhost:8222/',
4     timeout: 15000 // 请求超时时间
5 })
6
7 .....
8
9 // http response 拦截器
10 // 返回状态码改为 20000
11 service.interceptors.response.use(
12     response => {
13         if (response.data.code !== 20000) {
14
15 .....

```

