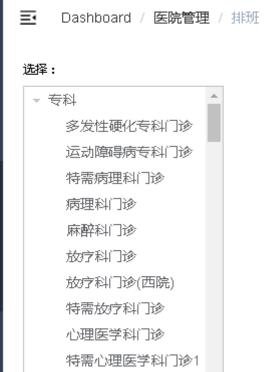
一、科室列表 (接口)





1、添加service接口和实现

(1) 在DepartmentService定义方法

```
1 //根据医院编号,查询医院所有科室列表
2 List<DepartmentVo> findDeptTree(String hoscode);
```

(2) 在DepartmentServiceImpl实现方法

```
1 //根据医院编号,查询医院所有科室列表
2 @Override
3 public List<DepartmentVo> findDeptTree(String hoscode) {
4     //创建list集合,用于最终数据封装
5     List<DepartmentVo> result = new ArrayList<>();
6     //根据医院编号,查询医院所有科室信息
7     Department departmentQuery = new Department();
9     departmentQuery.setHoscode(hoscode);
```

```
10
       Example example = Example.of(departmentQuery);
       //所有科室列表 departmentList
11
       List<Department> departmentList = departmentRepository.findAll(example);
12
13
       //根据大科室编号 bigcode 分组,获取每个大科室里面下级子科室
14
       Map<String, List<Department>> departmentMap =
15
16
              departmentList.stream().collect(Collectors.groupingBy(Department::getBigc
       //遍历map集合 deparmentMap
17
       for(Map.Entry<String,List<Department>> entry : departmentMap.entrySet()) {
18
          //大科室编号
19
          String bigcode = entry.getKey();
20
          //大科室编号对应的全局数据
21
           List<Department> department1List = entry.getValue();
22
          //封装大科室
23
          DepartmentVo departmentVo1 = new DepartmentVo();
24
           departmentVo1.setDepcode(bigcode);
25
26
           departmentVo1.setDepname(department1List.get(0).getBigname());
27
          //封装小科室
28
           List<DepartmentVo> children = new ArrayList<>();
29
30
           for(Department department: department1List) {
              DepartmentVo departmentVo2 = new DepartmentVo();
31
              departmentVo2.setDepcode(department.getDepcode());
32
33
              departmentVo2.setDepname(department.getDepname());
              //封装到list集合
34
              children.add(departmentVo2);
35
          }
36
          //把小科室list集合放到大科室children里面
37
           departmentVo1.setChildren(children);
38
          //放到最终result里面
39
           result.add(departmentVo1);
40
41
      }
       //返回
42
       return result;
43
44 }
```

2、添加DepartmentController方法

```
1 @RestController
2 @RequestMapping("/admin/hosp/department")
```

```
3 public class DepartmentController {
 4
 5
      @Autowired
       private DepartmentService departmentService;
 6
 7
      //根据医院编号,查询医院所有科室列表
 8
 9
      @ApiOperation(value = "查询医院所有科室列表")
      @GetMapping("getDeptList/{hoscode}")
10
      public R getDeptList(@PathVariable String hoscode) {
11
           List<DepartmentVo> list = departmentService.findDeptTree(hoscode);
12
          return R.ok().data("list",list);
13
14
      }
15 }
```

二、科室列表 (前端)

1、添加隐藏路由

```
1 {
2  path: 'hospital/schedule/:hoscode',
3  name: '排班',
4  component: () => import('@/views/hosp/schedule'),
5  meta: { title: '排班', noCache: true },
6  hidden: true
7 }
```

2、封装api方法

```
1 //查看医院科室
2 getDeptByHoscode(hoscode) {
3    return request ({
4        url: `/admin/hosp/department/getDeptList/${hoscode}`,
5        method: 'get'
6    })
7 },
```

3、添加/views/hosp/schedule.vue组件

```
1 <template>
       <div class="app-container">
 2
 3
           <div style="margin-bottom: 10px;font-size: 10px;">选择: </div>
 4
               <el-container style="height: 100%">
               <el-aside width="200px" style="border: 1px silver solid">
 5
                   <!-- 部门 -->
 6
 7
                   <el-tree
                    :data="data"
 8
                   :props="defaultProps"
9
                   :default-expand-all="true"
10
                   @node-click="handleNodeClick">
11
                   </el-tree>
12
               </el-aside>
13
               <el-main style="padding: 0 0 0 20px;">
14
                   <el-row style="width: 100%">
15
                   <!-- 排班日期 分页 -->
16
                   </el-row>
17
                   <el-row style="margin-top: 20px;">
18
                   <!-- 排班日期对应的排班医生 -->
19
                   </el-row>
20
21
               </el-main>
           </el-container>
22
       </div>
23
24 </template>
25 <script>
26 import hospApi from '@/api/yygh/hosp'
27 export default {
       data() {
28
29
           return {
30
               data: [],
               defaultProps: {
31
                   children: 'children',
32
                   label: 'depname'
33
34
               },
               hoscode: null
35
           }
36
37
       },
38
       created(){
39
           this.hoscode = this.$route.params.hoscode
```

```
40
           this.fetchData()
       },
41
       methods:{
42
           fetchData() {
43
               hospApi.getDeptByHoscode(this.hoscode)
44
45
                    .then(response => {
                       this.data = response.data.list
46
                   })
47
48
           }
49
       }
50 }
51 </script>
52 <style>
     .el-tree-node.is-current > .el-tree-node__content {
53
54
       background-color: #409EFF !important;
       color: white;
55
      }
56
57
     .el-checkbox__input.is-checked+.el-checkbox__label {
58
       color: black;
59
60
61 </style>
```

说明:底部style标签是为了控制树形展示数据选中效果的