

一、排班详情列表（接口）

选择：

▼ 专科

多发性硬化专科门诊

运动障碍病专科门诊

特需病理科门诊

病理科门诊

麻醉科门诊

放疗科门诊

放疗科门诊(西院)

特需放疗科门诊

心理医学科门诊

特需心理医学科门诊1

介入治疗门诊(放射科)

超声介入门诊

物理医学与康复科门诊

2020-12-21 周一

22 / 33

2020-12-30 周三

32 / 134

2021-01-01 周五

38 / 100

2021-01-02 周六

38 / 100

2021-01-03 周日

38 / 100

< 1 2 3 4 5 6 >

序号	职称	号源时间	可预约数	剩余预约数	挂号费(元)	擅长
1	副主任医师 裴育	上午	40	6	100	骨
2	副主任医师 邵迎红	下午	27	10	100	内
3	副主任医师 裴育	上午	40	6	100	骨
4	副主任医师 邵迎红	下午	27	10	100	内

1、添加service接口和实现

根据医院编号、科室编号和工作日期，查询排班详细信息

(1) 在ScheduleService定义方法

```
1 //根据医院编号、科室编号和工作日期，查询排班详细信息
2 List<Schedule> getDetailSchedule(String hoscode, String deocode, String workDate);
```

(2) 在ScheduleServiceImpl实现方法

```
1 //根据医院编号、科室编号和工作日期，查询排班详细信息
2 @Override
3 public List<Schedule> getDetailSchedule(String hoscode, String deocode, String workDate) {
4     //根据参数查询mongodb
5     List<Schedule> scheduleList =
6         scheduleRepository.findScheduleByHoscodeAndDeocodeAndWorkDate(hoscode, deocode, workDate);
7     //把得到list集合遍历，向设置其他值：医院名称、科室名称、日期对应星期
8     scheduleList.stream().forEach(item->{
9         this.packageSchedule(item);
10    });
11 }
```

```

11     return scheduleList;
12 }
13 //封装排班详情其他值 医院名称、科室名称、日期对应星期
14 private void packageSchedule(Schedule schedule) {
15     //设置医院名称
16     schedule.getParam().put("hosname",hospitalService.getHospName(schedule.getHoscode));
17     //设置科室名称
18     schedule.getParam().put("depname",
19         departmentService.getDepName(schedule.getHoscode(),schedule.getDepcode()));
20     //设置日期对应星期
21     schedule.getParam().put("dayOfWeek",this.getDayOfWeek(new DateTime(schedule.getWo
22 }

```

2、添加ScheduleRepository方法

```

1 //根据医院编号 、科室编号和工作日期，查询排班详细信息
2 List<Schedule> findScheduleByHoscodeAndDepcodeAndWorkDate(String hoscode, String depcode, String workDate) {

```

3、添加根据部门编码获取部门名称

(1) DepartmentService添加方法

```

1 //根据科室编号，和医院编号，查询科室名称
2 String getDepName(String hoscode, String depcode);

```

(2) DepartmentServiceImpl实现方法

```

1 //根据科室编号，和医院编号，查询科室名称
2 @Override
3 public String getDepName(String hoscode, String depcode) {
4     Department department = departmentRepository.getDepartmentByHoscodeAndDepcode(hoscode, depcode);
5     if(department != null) {
6         return department.getDepname();
7     }

```

```
8     return null;
9 }
```

2、在ScheduleController添加方法

```
1 //根据医院编号 、科室编号和工作日期，查询排班详细信息
2 @ApiOperation(value = "查询排班详细信息")
3 @GetMapping("getScheduleDetail/{hoscode}/{depcode}/{workDate}")
4 public R getScheduleDetail( @PathVariable String hoscode,
5                             @PathVariable String depcode,
6                             @PathVariable String workDate) {
7     List<Schedule> list = scheduleService.getDetailSchedule(hoscode,depcode,workDate)
8     return R.ok().data("list",list);
9 }
```

二、排班详情列表（前端）

1、封装api请求

在api/yygh/schedule.js添加

```
1 //查询排班详情
2 getScheduleDetail(hoscode,depcode,workDate) {
3     return request ({
4         url: `/admin/hosp/schedule/getScheduleDetail/${hoscode}/${depcode}/${workDate}`
5         method: 'get'
6     })
7 }
```

2、页面显示

修改/views/hosp/schedule.vue组件

```
1 <template>
2     <div class="app-container">
3         <div style="margin-bottom: 10px;font-size: 10px;">选择: </div>
```

```
4 <el-container style="height: 100%">
5 <el-aside width="200px" style="border: 1px silver solid">
6   <!-- 部门 -->
7   <el-tree
8     :data="data"
9     :props="defaultProps"
10    :default-expand-all="true"
11    @node-click="handleNodeClick">
12   </el-tree>
13 </el-aside>
14 <el-main style="padding: 0 0 0 20px;">
15   <el-row style="width: 100%">
16     <!-- 排班日期 分页 -->
17     <el-tag v-for="(item,index) in bookingScheduleList" :key="item.id"
18       {{ item.workDate }} {{ item.dayOfWeek }}<br/>
19       {{ item.availableNumber }} / {{ item.reservedNumber }}
20     </el-tag>
21
22     <!-- 分页 -->
23     <el-pagination
24       :current-page="page"
25       :total="total"
26       :page-size="limit"
27       class="pagination"
28       layout="prev, pager, next"
29       @current-change="getPage">
30     </el-pagination>
31
32   </el-row>
33   <el-row style="margin-top: 20px;">
34     <!-- 排班日期对应的排班医生 -->
35     <el-table
36       v-loading="listLoading"
37       :data="scheduleList"
38       border
39       fit
40       highlight-current-row>
41       <el-table-column
42         label="序号"
43         width="60"
44         align="center">
45         <template slot-scope="scope">
46           {{ scope.$index + 1 }}
```

```

47         </template>
48     </el-table-column>
49     <el-table-column label="职称" width="150">
50         <template slot-scope="scope">
51             {{ scope.row.title }} | {{ scope.row.docr
52         </template>
53     </el-table-column>
54     <el-table-column label="号源时间" width="80">
55         <template slot-scope="scope">
56             {{ scope.row.workTime == 0 ? "上午" : "下
57         </template>
58     </el-table-column>
59     <el-table-column prop="reservedNumber" label="可预约数" width
60     <el-table-column prop="availableNumber" label="剩余预约数" wi
61     <el-table-column prop="amount" label="挂号费(元)" width="90"/
62     <el-table-column prop="skill" label="擅长技能"/>
63 </el-table>
64 </el-row>
65 </el-main>
66 </el-container>
67 </div>
68 </template>
69 <script>
70 import hospApi from '@api/yygh/hosp'
71 import schApi from '@api/yygh/schedule'
72 export default {
73     data() {
74         return {
75             data: [],
76             defaultProps: {
77                 children: 'children',
78                 label: 'depname'
79             },
80             hoscode: null,
81             activeIndex: 0,
82             depcode: null,
83             depname: null,
84             workDate: null,
85
86             bookingScheduleList: [],
87             baseMap: {},
88
89             page: 1, // 当前页

```

```

90         limit: 7, // 每页个数
91         total: 0, // 总页码
92
93         scheduleList:[] //排班详情
94     },
95 },
96 created(){
97     this.hoscode = this.$route.params.hoscode
98     this.workDate = this.getCurDate()
99     this.fetchData()
100 },
101 methods:{
102     //查询排班详情
103     getDetailSchedule() {
104         schApi.getScheduleDetail(this.hoscode,this.decode,this.workDate)
105             .then(response => {
106                 this.scheduleList = response.data.list
107             })
108     },
109
110     fetchData() {
111         hospApi.getDeptByHoscode(this.hoscode)
112             .then(response => {
113                 this.data = response.data.list
114                 // 默认选中第一个
115                 if (this.data.length > 0) {
116                     this.decode = this.data[0].children[0].decode
117                     this.depname = this.data[0].children[0].depname
118
119                     this.getPage()
120                 }
121             })
122     },
123     getPage(page = 1) {
124         this.page = page
125         this.workDate = null
126         this.activeIndex = 0
127         this.getScheduleRule()
128     },
129
130     getScheduleRule() {
131         schApi.getScheduleRule(this.page, this.limit, this.hoscode, this.decode,
132             this.bookingScheduleList = response.data.bookingScheduleRuleList

```

```

133
134         this.total = response.data.total
135
136         this.scheduleList = response.data.scheduleList
137         this.baseMap = response.data.baseMap
138
139         // 分页后workDate=null, 默认选中第一个
140         if (this.workDate == null) {
141
142             this.workDate = this.bookingScheduleList[0].workDate
143         }
144         //调用查询排班详情
145         this.getDetailSchedule()
146     })
147 },
148
149     handleNodeClick(data) {
150         // 科室大类直接返回
151         if (data.children != null) return
152         this.depcode = data.depcode
153         this.depname = data.depname
154
155         this.getPage(1)
156     },
157
158     selectDate(workDate, index) {
159         this.workDate = workDate
160         this.activeIndex = index
161         //调用查询排班详情
162         this.getDetailSchedule()
163     },
164
165     getCurDate() {
166         var datetime = new Date()
167         var year = datetime.getFullYear()
168         var month = datetime.getMonth() + 1 < 10 ? '0' + (datetime.getMonth() + 1) : datetime.getMonth() + 1
169         var date = datetime.getDate() < 10 ? '0' + datetime.getDate() : datetime.getDate()
170         return year + '-' + month + '-' + date
171     }
172 }
173 }
174
175 </script>

```

```
176 <style>
177   .el-tree-node.is-current > .el-tree-node__content {
178     background-color: #409EFF !important;
179     color: white;
180   }
181
182   .el-checkbox__input.is-checked+.el-checkbox__label {
183     color: black;
184   }
185 </style>
```