

# 一、排班日期列表（接口）

Dashboard / 医院管理 / 排班

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

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
< 1 2 3 4 5 6 >			

根据医院编号 和 科室编号，查询排班规则数据

## 1、在ScheduleService添加方法和实现

### (1) ScheduleService定义方法

```
1 //根据医院编号 和 科室编号，查询排班规则数据
2 Map<String, Object> getRuleSchedule(long page, long limit, String hoscode, St
```

### (2) ScheduleServiceImpl实现方法

```
1 @Autowired
2 private ScheduleRepository scheduleRepository;
3
4 @Autowired
5 private MongoTemplate mongoTemplate;
6
7 @Autowired
8 private HospitalService hospitalService;
9
10 @Override
11 public Map<String, Object> getRuleSchedule(long page, long limit, String hoscode,
12 //1 根据医院编号 和 科室编号 查询
13 Criteria criteria = Criteria.where("hoscode").is(hoscode).and("depcode").
14
```

```

15 //2 根据工作日workDate期进行分组
16 Aggregation agg = Aggregation.newAggregation(
17     Aggregation.match(criteria), //匹配条件
18     Aggregation.group("workDate") //分组字段
19     .first("workDate").as("workDate")
20 //3 统计号源数量
21     .count().as("docCount")
22     .sum("reservedNumber").as("reservedNumber")
23     .sum("availableNumber").as("availableNumber"),
24 //排序
25     Aggregation.sort(Sort.Direction.ASC, "workDate"),
26 //4 实现分页
27     Aggregation.skip((page-1)*limit),
28     Aggregation.limit(limit)
29 );
30 //调用方法，最终执行
31 AggregationResults<BookingScheduleRuleVo> aggResults =
32     mongoTemplate.aggregate(agg, Schedule.class, BookingScheduleRuleVo.class);
33 List<BookingScheduleRuleVo> bookingScheduleRuleVoList = aggResults.getMappedResults();
34
35 //分组查询的总记录数
36 Aggregation totalAgg = Aggregation.newAggregation(
37     Aggregation.match(criteria),
38     Aggregation.group("workDate")
39 );
40 AggregationResults<BookingScheduleRuleVo> totalAggResults =
41     mongoTemplate.aggregate(totalAgg,
42         Schedule.class, BookingScheduleRuleVo.class);
43 int total = totalAggResults.getMappedResults().size();
44
45 //把日期对应星期获取
46 for(BookingScheduleRuleVo bookingScheduleRuleVo:bookingScheduleRuleVoList)
47     Date workDate = bookingScheduleRuleVo.getWorkDate();
48     String dayOfWeek = this.getDayOfWeek(new DateTime(workDate));
49     bookingScheduleRuleVo.setDayOfWeek(dayOfWeek);
50 }
51
52 //设置最终数据，进行返回
53 Map<String, Object> result = new HashMap<>();
54 result.put("bookingScheduleRuleList", bookingScheduleRuleVoList);
55 result.put("total", total);
56
57 //获取医院名称

```

```

58     String hosName = hospitalService.getHospName(hoscode);
59     //其他基础数据
60     Map<String, String> baseMap = new HashMap<>();
61     baseMap.put("hosname", hosName);
62     result.put("baseMap", baseMap);
63     return result;
64 }

```

### (3) 添加日期转换星期方法

```

1  ### 引入依赖
2  <dependency>
3      <groupId>joda-time</groupId>
4      <artifactId>joda-time</artifactId>
5  </dependency>
6
7  /**
8   * 根据日期获取周几数据
9   * @param dateTime
10  * @return
11  */
12 private String getDayOfWeek(DateTime dateTime) {
13     String dayOfWeek = "";
14     switch (dateTime.getDayOfWeek()) {
15         case DateTimeConstants.SUNDAY:
16             dayOfWeek = "周日";
17             break;
18         case DateTimeConstants.MONDAY:
19             dayOfWeek = "周一";
20             break;
21         case DateTimeConstants.TUESDAY:
22             dayOfWeek = "周二";
23             break;
24         case DateTimeConstants.WEDNESDAY:
25             dayOfWeek = "周三";
26             break;
27         case DateTimeConstants.THURSDAY:
28             dayOfWeek = "周四";
29             break;
30         case DateTimeConstants.FRIDAY:
31             dayOfWeek = "周五";

```

```
32         break;
33     case DateTimeConstants.SATURDAY:
34         dayOfWeek = "周六";
35     default:
36         break;
37 }
38 return dayOfWeek;
39 }
```

#### (4) HospService添加医院编号获取医院名称方法

```
1 //定义方法
2 String getHospName(String hoscode);
3
4 //实现方法
5 @Override
6 public String getHospName(String hoscode) {
7     Hospital hospital = hospitalRepository.getHospitalByHoscode(hoscode);
8     if(null != hospital) {
9         return hospital.getHosname();
10    }
11    return "";
12 }
```

## (5) ScheduleController添加方法

```
1 @RestController
2 @RequestMapping("/admin/hosp/schedule")
3 public class ScheduleController {
4
5     @Autowired
6     private ScheduleService scheduleService;
7
8     //根据医院编号 和 科室编号 ， 查询排班规则数据
9     @ApiOperation(value = "查询排班规则数据")
10    @GetMapping("getScheduleRule/{page}/{limit}/{hoscode}/{depcode}")
11    public R getScheduleRule(@PathVariable long page,
12                             @PathVariable long limit,
```

```

13         @PathVariable String hoscode,
14         @PathVariable String deocode) {
15     Map<String,Object> map
16         = scheduleService.getRuleSchedule(page,limit,hoscode,deocode);
17     return R.ok().data(map);
18 }
19
20 }

```

## 二、排班日期列表（前端）

### 1、封装api方法

在api/yygh创建schedule.js

```

1 getScheduleRule(page, limit, hoscode, deocode) {
2     return request({
3         url: `/admin/hosp/schedule/getScheduleRule/${page}/${limit}/${hoscode}/${deocode}`,
4         method: 'get'
5     })
6 }

```

### 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
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-row>
36         </el-main>
37     </el-container>
38 </div>
39 </template>
40 <script>
41 import hospApi from '@api/yygh/hosp'
42 import schApi from '@api/yygh/schedule'
43 export default {
44     data() {
45         return {
46             data: [],
47             defaultProps: {
48                 children: 'children',
49                 label: 'depname'
50             },
51             hoscode: null,
52             activeIndex: 0,
53             depcode: null,
54             depname: null,
55             workDate: null,
56
```

```
57         bookingScheduleList: [],
58         baseMap: {},
59
60         page: 1, // 当前页
61         limit: 7, // 每页个数
62         total: 0 // 总页码
63     }
64 },
65 created(){
66     this.hoscode = this.$route.params.hoscode
67     this.workDate = this.getCurDate()
68     this.fetchData()
69 },
70 methods:{
71     fetchData() {
72         hospApi.getDeptByHoscode(this.hoscode)
73             .then(response => {
74                 this.data = response.data.list
75                 // 默认选中第一个
76                 if (this.data.length > 0) {
77                     this.decode = this.data[0].children[0].decode
78                     this.depname = this.data[0].children[0].depname
79
80                     this.getPage()
81                 }
82             })
83     },
84     getPage(page = 1) {
85         this.page = page
86         this.workDate = null
87         this.activeIndex = 0
88         this.getScheduleRule()
89     },
90
91     getScheduleRule() {
92         schApi.getScheduleRule(this.page, this.limit, this.hoscode, this.dep
93             this.bookingScheduleList = response.data.bookingScheduleRule
94
95             this.total = response.data.total
96
97             this.scheduleList = response.data.scheduleList
98             this.baseMap = response.data.baseMap
99     }
```

```
100         // 分页后workDate=null, 默认选中第一个
101         if (this.workDate == null) {
102
103             this.workDate = this.bookingScheduleList[0].workDate
104         }
105     })
106 },
107
108     handleNodeClick(data) {
109         // 科室大类直接返回
110         if (data.children != null) return
111         this.depcode = data.depcode
112         this.depname = data.depname
113
114         this.getPage(1)
115     },
116
117     selectDate(workDate, index) {
118         this.workDate = workDate
119         this.activeIndex = index
120     },
121
122     getCurDate() {
123         var datetime = new Date()
124         var year = datetime.getFullYear()
125         var month = datetime.getMonth() + 1 < 10 ? '0' + (datetime.getMo
126         var date = datetime.getDate() < 10 ? '0' + datetime.getDate() :
127         return year + '-' + month + '-' + date
128     }
129 }
130 }
131
132 </script>
133 <style>
134     .el-tree-node.is-current > .el-tree-node__content {
135         background-color: #409EFF !important;
136         color: white;
137     }
138
139     .el-checkbox__input.is-checked+.el-checkbox__label {
140         color: black;
141     }
142 </style>
```



