

一、预约挂号详情

1、需求

北京协和医院 | 专科

多发性硬化专科门诊

2021年06月

2021-06-15 周二	2021-06-16 周三	2021-06-17 周四	2021-06-18 周五	2021-06-19 周六	2021-06-20 周日	2021-06-21 周一
停止挂号	有号	有号	有号	有号	有号	有号

< 1 2 >

上午号源

医师 | 邵迎红

¥ 100

剩余22

内分泌科常见病。

接口分析

- (1) 根据预约周期，展示可预约日期数据，按分页展示
- (2) 选择日期展示当天可预约列表（该接口后台已经实现过）

页面展示分析

- (1) 分页展示可预约日期，根据有号、无号、约满等状态展示不同颜色，以示区分
- (2) 可预约最后一个日期为即将放号日期，根据放号时间页面展示倒计时

2、预约挂号详情接口

(1) 在ScheduleService添加方法

```
1 /**
2     * 获取排班可预约日期数据
3     * @param page
4     * @param limit
5     * @param hoscode
```

```

6      * @param deocode
7      */
8 Map<String, Object> getBookingScheduleRule(Integer page, Integer limit, String hoscode

```

(2) 在ScheduleServiceImpl实现方法

```

1 //获取排班可预约日期数据
2 @Override
3 public Map<String, Object> getBookingScheduleRule(Integer page, Integer limit, String
4     Map<String, Object> result = new HashMap<>();
5
6     //获取预约规则
7     Hospital hospital = hospitalService.getByHoscode(hoscode);
8     if(null == hospital) {
9         throw new YyghException();
10    }
11    BookingRule bookingRule = hospital.getBookingRule();
12
13    //获取可预约日期分页数据
14    IPage iPage = this.getListDate(page, limit, bookingRule);
15    //当前页可预约日期
16    List<Date> dateList = iPage.getRecords();
17    //获取可预约日期科室剩余预约数
18    Criteria criteria = Criteria.where("hoscode").is(hoscode).and("deocode").is(depcod
19    Aggregation agg = Aggregation.newAggregation(
20        Aggregation.match(criteria),
21        Aggregation.group("workDate")//分组字段
22        .first("workDate").as("workDate")
23        .count().as("docCount")
24        .sum("availableNumber").as("availableNumber")
25        .sum("reservedNumber").as("reservedNumber")
26    );
27    AggregationResults<BookingScheduleRuleVo> aggregationResults = mongoTemplate.aggr
28    List<BookingScheduleRuleVo> scheduleVoList = aggregationResults.getMappedResults(
29    //获取科室剩余预约数
30
31    //合并数据 将统计数据ScheduleVo根据“安排日期”合并到BookingRuleVo
32    Map<Date, BookingScheduleRuleVo> scheduleVoMap = new HashMap<>();
33    if(!CollectionUtils.isEmpty(scheduleVoList)) {
34        scheduleVoMap = scheduleVoList.stream().collect(Collectors.toMap(BookingSched

```

```
35     }
36     //获取可预约排班规则
37     List<BookingScheduleRuleVo> bookingScheduleRuleVoList = new ArrayList<>();
38     for(int i=0, len=dateList.size(); i<len; i++) {
39         Date date = dateList.get(i);
40
41         BookingScheduleRuleVo bookingScheduleRuleVo = scheduleVoMap.get(date);
42         if(null == bookingScheduleRuleVo) { // 说明当天没有排班医生
43             bookingScheduleRuleVo = new BookingScheduleRuleVo();
44             //就诊医生人数
45             bookingScheduleRuleVo.setDocCount(0);
46             //科室剩余预约数 -1表示无号
47             bookingScheduleRuleVo.setAvailableNumber(-1);
48         }
49         bookingScheduleRuleVo.setWorkDate(date);
50         bookingScheduleRuleVo.setWorkDateMd(date);
51         //计算当前预约日期为周几
52         String dayOfWeek = this.getDayOfWeek(new DateTime(date));
53         bookingScheduleRuleVo.setDayOfWeek(dayOfWeek);
54
55         //最后一页最后一条记录为即将预约 状态 0: 正常 1: 即将放号 -1: 当天已停止挂号
56         if(i == len-1 && page == iPage.getPages()) {
57             bookingScheduleRuleVo.setStatus(1);
58         } else {
59             bookingScheduleRuleVo.setStatus(0);
60         }
61         //当天预约如果过了停号时间, 不能预约
62         if(i == 0 && page == 1) {
63             DateTime stopTime = this.getDateTime(new Date(), bookingRule.getStopTime(
64                 if(stopTime.isBeforeNow()) {
65                     //停止预约
66                     bookingScheduleRuleVo.setStatus(-1);
67                 }
68             }
69             bookingScheduleRuleVoList.add(bookingScheduleRuleVo);
70         }
71
72         //可预约日期规则数据
73         result.put("bookingScheduleList", bookingScheduleRuleVoList);
74         result.put("total", iPage.getTotal());
75         //其他基础数据
76         Map<String, String> baseMap = new HashMap<>();
77         //医院名称
```

```

78     baseMap.put("hosname", hospitalService.getHospName(hoscode));
79     //科室
80     Department department = departmentService.getDepartment(hoscode, deptime);
81     //大科室名称
82     baseMap.put("bigname", department.getBigname());
83     //科室名称
84     baseMap.put("depname", department.getDepname());
85     //月
86     baseMap.put("workDateString", new DateTime().toString("yyyy年MM月"));
87     //放号时间
88     baseMap.put("releaseTime", bookingRule.getReleaseTime());
89     //停号时间
90     baseMap.put("stopTime", bookingRule.getStopTime());
91     result.put("baseMap", baseMap);
92     return result;
93
94 }

```

(3) 添加获取可预约日期分页数据方法

```

1  /**
2   * 获取可预约日期分页数据
3   */
4  private IPage<Date> getListDate(int page, int limit, BookingRule bookingRule) {
5      //当天放号时间
6      DateTime releaseTime = this.getDateTime(new Date(), bookingRule.getReleaseTime());
7      //预约周期
8      int cycle = bookingRule.getCycle();
9      //如果当天放号时间已过，则预约周期后一天为即将放号时间，周期加1
10     if(releaseTime.isBeforeNow()) cycle += 1;
11     //可预约所有日期，最后一天显示即将放号倒计时
12     List<Date> dateList = new ArrayList<>();
13     for (int i = 0; i < cycle; i++) {
14         //计算当前预约日期
15         DateTime curDateTime = new DateTime().plusDays(i);
16         String dateString = curDateTime.toString("yyyy-MM-dd");
17         dateList.add(new DateTime(dateString).toDate());
18     }
19     //日期分页，由于预约周期不一样，页面一排最多显示7天数据，多了就要分页显示
20     List<Date> pageDateList = new ArrayList<>();

```

```

21     int start = (page-1)*limit;
22     int end = (page-1)*limit+limit;
23     if(end > dateList.size()) end = dateList.size();
24     for (int i = start; i < end; i++) {
25         pageDateList.add(dateList.get(i));
26     }
27
28     IPage<Date> iPage = new com.baomidou.mybatisplus.extension.plugins.pagination
29     iPage.setRecords(pageDateList);
30     return iPage;
31 }

```

(4) 将Date日期 (yyyy-MM-dd HH:mm) 转换为DateTime

```

1  /**
2   * 将Date日期 (yyyy-MM-dd HH:mm) 转换为DateTime
3   */
4  private DateTime getDateTime(Date date, String timeString) {
5      String dateTimeString = new DateTime(date).toString("yyyy-MM-dd") + " " + timeString;
6      DateTime dateTime = DateTimeFormat.forPattern("yyyy-MM-dd HH:mm").parseDateTime(dateTimeString);
7      return dateTime;
8  }

```

(5) 在DepartmentService类添加方法和实现

```

1  /**
2   * 根据医院编号 和 科室编号获取科室数据
3   */
4  Department getDepartment(String hoscode, String deocode);
5
6  //实现方法：根据医院编号 和 科室编号获取科室数据
7  @Override
8  public Department getDepartment(String hoscode, String deocode) {
9      return departmentRepository.getDepartmentByHoscodeAndDeocode(hoscode, deocode);
10 }

```

(6) 在HospitalApiController添加方法

```
1 @Autowired
2 private ScheduleService scheduleService;
3
4 @ApiOperation(value = "获取可预约排班数据")
5 @GetMapping("auth/getBookingScheduleRule/{page}/{limit}/{hoscode}/{depcode}")
6 public R getBookingSchedule(
7     @PathVariable Integer page,
8     @PathVariable Integer limit,
9     @PathVariable String hoscode,
10    @PathVariable String depcode) {
11
12    Map<String, Object> map = scheduleService.getBookingScheduleRule(page, limit, hoscode, depcode);
13    return R.ok().data(map);
14 }
15
16 @ApiOperation(value = "获取排班数据")
17 @GetMapping("auth/findScheduleList/{hoscode}/{depcode}/{workDate}")
18 public R findScheduleList(
19     @PathVariable String hoscode,
20     @PathVariable String depcode,
21     @PathVariable String workDate) {
22    List<Schedule> scheduleList = scheduleService.getDetailShedule(hoscode, depcode, workDate);
23    return R.ok().data("scheduleList", scheduleList);
24 }
```

3、预约挂号详情前端

(1) 在/api/hosp.js添加方法

```
1 getBookingScheduleRule(page, limit, hoscode, depcode) {
2     return request({
3         url: `${api_name}/auth/getBookingScheduleRule/${page}/${limit}/${hoscode}/${depcode}`,
4         method: 'get'
5     })
6 },
7
```

```

8 findScheduleList(hoscode, deocode, workDate) {
9     return request({
10         url: `${api_name}/auth/findScheduleList/${hoscode}/${deocode}/${workDate}`,
11         method: 'get'
12     })
13 },

```

(2) 页面显示

创建/pages/hospital/schedule.vue组件

```

1 <template>
2   <!-- header -->
3   <div class="nav-container page-component">
4     <!-- 左侧导航 #start -->
5     <div class="nav left-nav">
6       <div class="nav-item selected">
7         <span class="v-link selected dark" :onclick="'javascript:window.location=\'/'"
8       </div>
9       <div class="nav-item ">
10        <span class="v-link clickable dark" :onclick="'javascript:window.location=\'/'"
11      </div>
12      <div class="nav-item">
13        <span class="v-link clickable dark" :onclick="'javascript:window.location=\'/'"
14      </div>
15      <div class="nav-item "><span
16        class="v-link clickable dark"> 停诊信息 </span>
17      </div>
18      <div class="nav-item "><span
19        class="v-link clickable dark"> 查询/取消 </span>
20      </div>
21    </div>
22    <!-- 左侧导航 #end -->
23    <!-- 右侧内容 #start -->
24    <div class="page-container">
25      <div class="hospital-source-list">
26        <div class="header-wrapper" style="justify-content:normal">
27          <span class="v-link clickable" @click="show()">{{ baseMap.hosname}}</span>
28          <div class="split"></div>
29          <div>{{ baseMap.bigname }}</div>
30        </div>

```

```

31 <div class="title mt20"> {{ baseMap.depname }}</div>
32 <!-- 号源列表 #start -->
33 <div class="mt60">
34   <div class="title-wrapper">{{ baseMap.workDateString }}</div>
35   <div class="calendar-list-wrapper">
36     <!-- item.depNumber == -1 ? 'gray space' : item.depNumber == 0 ? 'gray'
37     <!-- selected , index == activeIndex ? 'selected' : ''-->
38     <div :class="'calendar-item '+item.curClass" style="width: 124px;"
39       v-for="(item, index) in bookingScheduleList" :key="item.id"
40       @click="selectDate(item, index)">
41       <div class="date-wrapper"><span>{{ item.workDate }}</span><span class="
42       <div class="status-wrapper" v-if="item.status == 0">{{ item.availableNu
43       <div class="status-wrapper" v-if="item.status == 1">即将放号</div>
44       <div class="status-wrapper" v-if="item.status == -1">停止挂号</div>
45     </div>
46   </div>
47   <!-- 分页 -->
48   <el-pagination
49     class="pagination"
50     layout="prev, pager, next"
51     :current-page="page"
52     :total="total"
53     :page-size="limit"
54     @current-change="getPage">
55   </el-pagination>
56 </div>
57 <!-- 即将放号 #start-->
58 <div class="countdown-wrapper mt60" v-if="!tabShow">
59   <div class="countdonw-title"> {{ time }}<span class="v-link selected">{{ ba
60   <div class="countdown-text"> 倒 计 时
61     <div>
62       <span class="number">{{ timeString }}</span>
63     </div>
64   </div>
65 </div>
66 <!-- 即将放号 #end-->
67 <!-- 号源列表 #end -->
68 <!-- 上午号源 #start -->
69 <div class="mt60" v-if="tabShow">
70   <div class="">
71     <div class="list-title">
72       <div class="block"></div>
73     上午号源

```



```

74     </div>
75     <div v-for="item in scheduleList" :key="item.id" >
76         <div class="list-item" v-if="item.workTime == 0">
77             <div class="item-wrapper">
78                 <div class="title-wrapper">
79                     <div class="title">{{ item.title }}</div>
80                     <div class="split"></div>
81                     <div class="name"> {{ item.docname }}</div>
82                 </div>
83                 <div class="special-wrapper">{{ item.skill }}</div>
84             </div>
85             <div class="right-wrapper">
86                 <div class="fee"> ¥{{ item.amount }}
87             </div>
88             <div class="button-wrapper">
89                 <div class="v-button" @click="booking(item.id, item.availableNumb
90                     <span>剩余<span class="number">{{ item.availableNumber }}</span>
91                 </div>
92             </div>
93         </div>
94     </div>
95 </div>
96 </div>
97 <!-- 上午号源 #end -->
98 <!-- 下午号源 #start -->
99 <div class="mt60" v-if="tabShow">
100     <div class="">
101         <div class="list-title">
102             <div class="block"></div>
103             下午号源
104         </div>
105         <div v-for="item in scheduleList" :key="item.id">
106             <div class="list-item" v-if="item.workTime == 1">
107                 <div class="item-wrapper">
108                     <div class="title-wrapper">
109                         <div class="title">{{ item.title }}</div>
110                         <div class="split"></div>
111                         <div class="name"> {{ item.docname }}</div>
112                     </div>
113                     <div class="special-wrapper">{{ item.skill }}</div>
114                 </div>
115                 <div class="right-wrapper">
116                     <div class="fee"> ¥{{ item.amount }}

```

```
117         </div>
118         <div class="button-wrapper">
119             <div class="v-button" @click="booking(item.id, item.availableNumt
120                 <span>剩余<span class="number">{{ item.availableNumber }}</span
121             </div>
122         </div>
123     </div>
124 </div>
125 </div>
126 </div>
127     <!-- 下午号源 #end -->
128 </div>
129 </div>
130     <!-- 右侧内容 #end -->
131 </div>
132 <!-- footer -->
133 </template>
134 <script>
135 import '~/assets/css/hospital_personal.css'
136 import '~/assets/css/hospital.css'
137
138 import hospitalApi from '@api/hosp'
139 export default {
140     data() {
141         return {
142             hoscode: null,
143             deocode: null,
144             workDate: null,
145             bookingScheduleList: [],
146             scheduleList : [],
147             baseMap : {},
148             nextWorkDate: null, // 下一页第一个日期
149             preWorkDate: null, // 上一页第一个日期
150             tabShow: true, // 挂号列表与即将挂号切换
151             activeIndex: 0,
152
153             page: 1, // 当前页
154             limit: 7, // 每页个数
155             total: 1, // 总页码
156
157             timeString: null,
158             time: '今天',
159             timer: null,
```

```
160
161     pageFirstStatus: 0 // 第一页第一条数据状态
162 }
163 },
164
165 created() {
166     this.hoscode = this.$route.query.hoscode
167     this.depcode = this.$route.query.depcode
168     this.workDate = this.getCurDate()
169     this.getBookingScheduleRule()
170 },
171
172 methods: {
173     getPage(page = 1) {
174         this.page = page
175         this.workDate = null
176         this.activeIndex = 0
177
178         this.getBookingScheduleRule()
179     },
180
181     getBookingScheduleRule() {
182         hospitalApi.getBookingScheduleRule(this.page, this.limit, this.hoscode, this.de
183             this.bookingScheduleList = response.data.bookingScheduleList
184             this.total = response.data.total
185             this.baseMap = response.data.baseMap
186
187             this.dealClass()
188
189             // 分页后workDate=null, 默认选中第一个
190             if (this.workDate == null) {
191                 this.workDate = this.bookingScheduleList[0].workDate
192             }
193             //判断当天是否停止预约 status == -1 停止预约
194             if(this.workDate == this.getCurDate()) {
195                 this.pageFirstStatus = this.bookingScheduleList[0].status
196             } else {
197                 this.pageFirstStatus = 0
198             }
199             this.findScheduleList()
200         })
201     },
202
```

```

203 findScheduleList() {
204     hospitalApi.findScheduleList(this.hoscode, this.decode, this.workDate).then(res => {
205         this.scheduleList = response.data.scheduleList
206     })
207 },
208
209 selectDate(item, index) {
210     this.workDate = item.workDate
211     this.activeIndex = index
212
213     //清理定时
214     if(this.timer != null) clearInterval(this.timer)
215
216     // 是否即将放号
217     if(item.status == 1) {
218         this.tabShow = false
219         // 放号时间
220         let releaseTime = new Date(this.getCurDate() + ' ' + this.baseMap.releaseTime)
221         let nowTime = new Date().getTime();
222         this.countDown(releaseTime, nowTime)
223
224         this.dealClass();
225     } else {
226         this.tabShow = true
227
228         this.getBookingScheduleRule()
229     }
230 },
231
232 dealClass() {
233     //处理样式
234     for (let i = 0; i < this.bookingScheduleList.length; i++) {
235         // depNumber -1:无号 0: 约满 >0: 有号
236         let curClass = this.bookingScheduleList[i].availableNumber == -1 ? 'gray space' : ''
237         curClass += i == this.activeIndex ? ' selected' : ''
238         this.bookingScheduleList[i].curClass = curClass
239     }
240 },
241
242 getCurDate() {
243     let datetime = new Date()
244     let year = datetime.getFullYear()
245     let month = datetime.getMonth() + 1 < 10 ? '0' + (datetime.getMonth() + 1) : datetime.getMonth() + 1

```

```

246     let date = datetime.getDate() < 10 ? '0' + datetime.getDate() : datetime.getDat
247     return year + '-' + month + '-' + date
248 },
249
250 countdown(releaseTime, nowTime) {
251     //计算倒计时时长
252     let secondes = 0;
253     if(releaseTime > nowTime) {
254         this.time = '今天'
255         //当前时间到放号时间的时长
256         secondes = Math.floor((releaseTime - nowTime) / 1000);
257     } else {
258         this.time = '明天'
259         //计算明天放号时间
260         let releaseDate = new Date(releaseTime)
261         releaseTime = new Date(releaseDate.setDate(releaseDate.getDate() + 1)).getTir
262         //当前时间到明天放号时间的时长
263         secondes = Math.floor((releaseTime - nowTime) / 1000);
264     }
265
266     //定时任务
267     this.timer = setInterval(() => {
268         secondes = secondes - 1
269         if(secondes <= 0) {
270             clearInterval(timer);
271             this.init()
272         }
273         this.timeString = this.convertTimeString(secondes)
274     }, 1000);
275     // 通过$once来监听定时器，在beforeDestroy钩子可以被清除。
276     this.$once('hook:beforeDestroy', () => {
277         clearInterval(timer);
278     })
279 },
280
281 convertTimeString(allseconds) {
282     if(allseconds <= 0) return '00:00:00'
283     // 计算天数
284     let days = Math.floor(allseconds / (60 * 60 * 24));
285     // 小时
286     let hours = Math.floor((allseconds - (days * 60 * 60 * 24)) / (60 * 60));
287     // 分钟
288     let minutes = Math.floor((allseconds - (days * 60 * 60 * 24) - (hours * 60 * 60

```

```
289 // 秒
290 let seconds = allseconds - (days * 60 * 60 * 24) - (hours * 60 * 60) - (minutes
291
292 //拼接时间
293 let timString = "";
294 if (days > 0) {
295     timString = days + "天:";
296 }
297 return timString += hours + " 时 " + minutes + " 分 " + seconds + " 秒 ";
298 },
299
300 show() {
301     window.location.href = '/hospital/' + this.hoscode
302 },
303
304 booking(scheduleId, availableNumber) {
305     if(availableNumber == 0 || this.pageFirstStatus == -1) {
306         this.$message.error('不能预约')
307     } else {
308         window.location.href = '/hospital/booking?scheduleId=' + scheduleId
309     }
310 }
311 }
312 }
313 </script>
```