

# 一、后台管理系统-预约统计功能

我们统计医院每天的预约情况，通过图表的形式展示，统计的数据都来自订单模块，因此我们在该模块封装好数据，在统计模块通过feign的形式获取数据。

我们为什么需要一个统计模块呢，因为在实际的生产环境中，有很多种各式统计，数据来源于各个服务模块，我们得有一个统计模块来专门管理

## 1、开发每天预约数据接口

操作模块：service\_orders

### (1) 在OrderInfoMapper添加方法

```
1 public interface OrderMapper extends BaseMapper<OrderInfo> {  
2  
3     //统计每天平台预约数据  
4     List<OrderCountVo> selectOrderCount(OrderCountQueryVo orderCountQueryVo);  
5 }
```

### (2) 在OrderInfoMapper.xml添加方法

创建OrderInfoMapper.xml文件

```
1 <?xml version="1.0" encoding="UTF-8" ?>  
2 <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"  
3     "http://mybatis.org/dtd/mybatis-3-mapper.dtd">  
4 <mapper namespace="com.atguigu.yygh.orders.mapper.OrderMapper">  
5  
6     <select id="selectOrderCount" resultType="com.atguigu.yygh.vo.order.OrderCountVo">  
7         select reserve_date as reserveDate, count(reserve_date) as count  
8         from order_info  
9         <where>  
10             <if test="hosname != null and hosname != ''">  
11                 and hosname like CONCAT('%',{hosname},'%')  
12             </if>  
13             <if test="reserveDateBegin != null and reserveDateBegin != ''">
```

```

14         and reserve_date >= #{reserveDateBegin}
15     </if>
16     <if test="reserveDateEnd != null and reserveDateEnd != ''">
17         and reserve_date <= #{reserveDateEnd}
18     </if>
19     and is_deleted = 0
20 </where>
21 group by reserve_date
22 order by reserve_date
23 </select>
24 </mapper>

```

### (3) 添加配置

#### application.properties添加

```

1 mybatis-plus.mapper-locations=classpath:com/atguigu/yygh/orders/mapper/xml/*.xml

```

#### service模块的pom.xml添加

```

1 <build>
2     <plugins>
3         <plugin>
4             <groupId>org.springframework.boot</groupId>
5             <artifactId>spring-boot-maven-plugin</artifactId>
6         </plugin>
7     </plugins>
8     <resources>
9         <resource>
10             <directory>src/main/java</directory>
11             <includes>
12                 <include>**/*.yml</include>
13                 <include>**/*.properties</include>
14                 <include>**/*.xml</include>
15             </includes>
16             <filtering>>false</filtering>
17         </resource>
18         <resource>

```

```

19         <directory>src/main/resources</directory>
20         <includes> <include>**/*.yml</include>
21             <include>**/*.properties</include>
22             <include>**/*.xml</include>
23         </includes>
24         <filtering>>false</filtering>
25     </resource>
26 </resources>
27 </build>

```

#### (4) OrderService添加方法和实现

```

1  /**
2      * 订单统计
3      */
4  Map<String, Object> getCountMap(OrderCountQueryVo orderCountQueryVo);
5
6  //实现方法
7  @Override
8  public Map<String, Object> getCountMap(OrderCountQueryVo orderCountQueryVo) {
9      Map<String, Object> map = new HashMap<>();
10
11      List<OrderCountVo> orderCountVoList
12          = baseMapper.selectOrderCount(orderCountQueryVo);
13      //日期列表
14      List<String> dateList
15          =orderCountVoList.stream().map(OrderCountVo::getReserveDate).collect(Collectors.toList());
16      //统计列表
17      List<Integer> countList
18          =orderCountVoList.stream().map(OrderCountVo::getCount).collect(Collectors.toList());
19      map.put("dateList", dateList);
20      map.put("countList", countList);
21      return map;
22  }

```

#### (5) 在OrderApiController添加方法

```

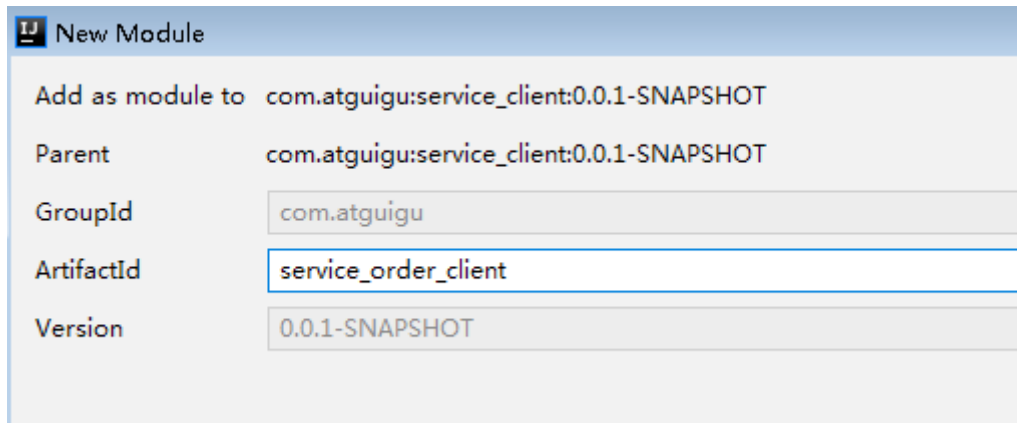
1  @ApiOperation(value = "获取订单统计数据")

```

```
2 @PostMapping("inner/getCountMap")
3 public Map<String, Object> getCountMap(@RequestBody OrderCountQueryVo orderCountQuery
4     return orderService.getCountMap(orderCountQueryVo);
5 }
```

## 2、封装远程调用接口

### (1) 创建模块：service\_order\_client




New Module	
Add as module to	com.atguigu:service_client:0.0.1-SNAPSHOT
Parent	com.atguigu:service_client:0.0.1-SNAPSHOT
GroupId	com.atguigu
ArtifactId	service_order_client
Version	0.0.1-SNAPSHOT

### (2) 添加feign接口

```
1 @FeignClient(value = "service-orders")
2 @Repository
3 public interface OrderFeignClient {
4     /**
5      * 获取订单统计数据
6      */
7     @PostMapping("/api/order/orderInfo/inner/getCountMap")
8     Map<String, Object> getCountMap(@RequestBody OrderCountQueryVo orderCountQueryVo)
9
10 }
```

## 3、搭建统计分析模块

### (1) 创建service\_statistics模块

 New Module

Add as module to	com.atguigu:service:0.0.1-SNAPSHOT
Parent	com.atguigu:service:0.0.1-SNAPSHOT
GroupId	<input type="text" value="com.atguigu"/>
ArtifactId	<input type="text" value="service_statistics"/>
Version	<input type="text" value="0.0.1-SNAPSHOT"/>

## (2) 引入依赖

```
1 <dependencies>
2   <dependency>
3     <groupId>com.atguigu</groupId>
4     <artifactId>service_order_client</artifactId>
5     <version>0.0.1-SNAPSHOT</version>
6   </dependency>
7 </dependencies>
```

## (3) 添加配置文件

```
1 # 服务端口
2 server.port=8209
3 # 服务名
4 spring.application.name=service-sta
5 # 环境设置: dev、test、prod
6 spring.profiles.active=dev
7
8 # nacos服务地址
9 spring.cloud.nacos.discovery.server-addr=127.0.0.1:8848
```

## (4) 添加启动类

```
1 @SpringBootApplication(exclude = DataSourceAutoConfiguration.class)
2 @EnableDiscoveryClient
3 @EnableFeignClients(basePackages = {"com.atguigu"})
```

```

4 @ComponentScan(basePackages = {"com.atguigu"})
5 public class ServiceStatisticsApplication {
6
7     public static void main(String[] args) {
8         SpringApplication.run(ServiceStatisticsApplication.class, args);
9     }
10
11 }

```

## (5) 添加controller方法

```

1 @Api(tags = "统计管理接口")
2 @RestController
3 @RequestMapping("/admin/statistics")
4 public class StatisticsController {
5
6     @Autowired
7     private OrderFeignClient orderFeignClient;
8
9     @ApiOperation(value = "获取订单统计数据")
10    @GetMapping("getCountMap")
11    public R getCountMap(@ApiParam(name = "orderCountQueryVo", value = "查询对象", required = true) OrderCountQueryVo orderCountQueryVo) {
12        Map<String, Object> map = orderFeignClient.getCountMap(orderCountQueryVo);
13        return R.ok().data(map);
14    }
15 }

```

## 4、整合统计功能前端

ECharts是百度的一个项目，后来百度把Echart捐给apache，用于图表展示，提供了常规的[折线图](#)、[柱状图](#)、[散点图](#)、[饼图](#)、[K线图](#)，用于统计的[盒形图](#)，用于地理数据可视化的[地图](#)、[热力图](#)、[线图](#)，用于关系数据可视化的[关系图](#)、[treemap](#)、[旭日图](#)，多维数据可视化的[平行坐标](#)，还有用于 BI 的[漏斗图](#)，[仪表盘](#)，并且支持图与图之间的混搭。

官方网站: <https://echarts.apache.org/zh/index.html>

### (1) 项目中安装 echarts组件

```
npm install --save echarts@4.1.0
```

## (2) 添加路由

```
1  {
2    path: '/statistics',
3    component: Layout,
4    redirect: '/statistics/order/index',
5    name: 'BasesInfo',
6    meta: { title: '统计管理', icon: 'table' },
7    alwaysShow: true,
8    children: [
9      {
10     path: 'order/index',
11     name: '预约统计',
12     component: () => import('@views/statistics/order/index'),
13     meta: { title: '预约统计' }
14   }
15 ]
16 },
```

## (3) 封装api请求

```
1  import request from '@utils/request'
2
3  const api_name = '/admin/statistics'
4
5  export default {
6
7    getCountMap(searchObj) {
8      return request({
9        url: `${api_name}/getCountMap`,
10       method: 'get',
11       params: searchObj
12     })
13   }
14 }
```

## (4) 添加组件

创建/views/statistics/order/index.vue组件

```
1 <template>
2   <div class="app-container">
3     <!-- 表单 -->
4     <el-form :inline="true" class="demo-form-inline">
5       <el-form-item>
6         <el-input v-model="searchObj.hosname" placeholder="点击输入医院名称"/>
7       </el-form-item>
8
9       <el-form-item>
10        <el-date-picker
11          v-model="searchObj.reserveDateBegin"
12          type="date"
13          placeholder="选择开始日期"
14          value-format="yyyy-MM-dd"/>
15        </el-form-item>
16        <el-form-item>
17          <el-date-picker
18            v-model="searchObj.reserveDateEnd"
19            type="date"
20            placeholder="选择截止日期"
21            value-format="yyyy-MM-dd"/>
22          </el-form-item>
23          <el-button
24            :disabled="btnDisabled"
25            type="primary"
26            icon="el-icon-search"
27            @click="showChart()">查询</el-button>
28        </el-form>
29
30        <div class="chart-container">
31          <div id="chart" ref="chart"
32            class="chart" style="height:500px;width:100%"/>
33        </div>
34      </div>
35 </template>
36
37 <script>
38 import echarts from 'echarts'
```



```
39 import statisticsApi from '@api/yygh/sta'
40
41 export default {
42
43   data() {
44     return {
45       searchObj: {
46         hosname: '',
47         reserveDateBegin: '',
48         reserveDateEnd: ''
49       },
50       btnDisabled: false,
51       chart: null,
52       title: '',
53       xData: [], // x轴数据
54       yData: [] // y轴数据
55     }
56   },
57
58   methods: {
59     // 初始化图表数据
60     showChart() {
61       statisticsApi.getCountMap(this.searchObj).then(response => {
62         this.yData = response.data.countList
63         this.xData = response.data.dateList
64         this.setChartData()
65       })
66     },
67
68     setChartData() {
69       // 基于准备好的dom，初始化echarts实例
70       var myChart = echarts.init(document.getElementById('chart'))
71       // 指定图表的配置项和数据
72       var option = {
73         title: {
74           text: this.title + '挂号量统计'
75         },
76         tooltip: {},
77         legend: {
78           data: [this.title]
79         },
80         xAxis: {
81           data: this.xData
```

```
82         },
83         yAxis: {
84             minInterval: 1
85         },
86         series: [{
87             name: this.title,
88             type: 'line',
89             data: this.yData
90         }]
91     }
92
93     // 使用刚指定的配置项和数据显示图表。
94     myChart.setOption(option)
95 },
96 }
97 }
98 </script>
```