尚医通

版本：V1.0

# 就医提醒

我们通过定时任务，每天8点执行，提醒就诊

## 1 、搭建定时任务模块service-task

### 1.1 搭建service-task服务

搭建方式如service-user

### 1.2 修改配置pom.xml

|  |
| --- |
| <**dependencies**>  <**dependency**>  <**groupId**>com.atguigu</**groupId**>  <**artifactId**>rabbit\_util</**artifactId**>  <**version**>0.0.1-SNAPSHOT</**version**>  </**dependency**>  </**dependencies**> |

说明：引入依赖

### 1.3 添加配置文件

1、applicationproperties

|  |
| --- |
| *# 服务端口*  **server.port**=**8207**  *# 服务名*  **spring.application.name**=**service-task**  *# 环境设置：dev、test、prod*  **spring.profiles.active**=**dev**  *# nacos服务地址*  **spring.cloud.nacos.discovery.server-addr**=**127.0.0.1:8848**  *#rabbitmq地址*  **spring.rabbitmq.host**=**192.168.44.165**  **spring.rabbitmq.port**=**5672**  **spring.rabbitmq.username**=**guest**  **spring.rabbitmq.password**=**guest** |

### 1.4 添加启动类

|  |
| --- |
| @SpringBootApplication(exclude = DataSourceAutoConfiguration.**class**)*//取消数据源自动配置*  @EnableDiscoveryClient  **public class** ServiceTaskApplication {  **public static void** main(String[] args) {  SpringApplication.*run*(ServiceTaskApplication.**class**, args);  }  } |

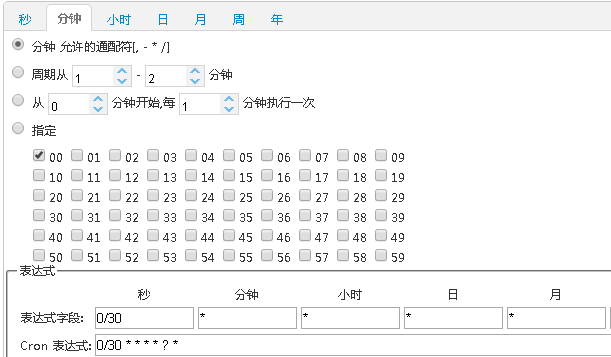
### 1.5 添加常量配置

在rabbit-util模块com.atguigu.yygh.common.constant.MqConst类添加

|  |
| --- |
| **public static final** String ***EXCHANGE\_DIRECT\_TASK*** = **"exchange.direct.task"**;  **public static final** String ***ROUTING\_TASK\_8*** = **"task.8"**;  *//队列*  **public static final** String ***QUEUE\_TASK\_8*** = **"queue.task.8"**; |

### 1.6 添加定时任务

Cron表达式



|  |
| --- |
| @Component  @EnableScheduling  **public class** ScheduledTask {  @Autowired  **private** RabbitService **rabbitService**;  */\*\**  *\* 每天8点执行 提醒就诊*  *\*/*  *//@Scheduled(cron = "0 0 1 \* \* ?")*  @Scheduled(cron = **"0/30 \* \* \* \* ?"**)  **public void** task1() {  **rabbitService**.sendMessage(MqConst.***EXCHANGE\_DIRECT\_TASK***, MqConst.***ROUTING\_TASK\_8***, **""**);  }  } |

## 2、添加就医提醒处理

操作模块service-order

### 2.1 添加service接口

在OrderService类添加接口

|  |
| --- |
| */\*\**  *\* 就诊提醒*  *\*/*  **void** patientTips(); |

### 2.2 添加service接口实现类

在OrderServiceImpl类添加接口实现

|  |
| --- |
| @Override  **public void** patientTips() {  QueryWrapper<OrderInfo> queryWrapper = **new** QueryWrapper<>();  queryWrapper.eq(**"reserve\_date"**,**new** DateTime().toString(**"yyyy-MM-dd"**));  List<OrderInfo> orderInfoList = **baseMapper**.selectList(queryWrapper);  **for**(OrderInfo orderInfo : orderInfoList) {  *//短信提示*  MsmVo msmVo = **new** MsmVo();  msmVo.setPhone(orderInfo.getPatientPhone());  String reserveDate = **new** DateTime(orderInfo.getReserveDate()).toString(**"yyyy-MM-dd"**) + (orderInfo.getReserveTime()==0 ? **"上午"**: **"下午"**);  Map<String,Object> param = **new** HashMap<String,Object>(){{  put(**"title"**, orderInfo.getHosname()+**"|"**+orderInfo.getDepname()+**"|"**+orderInfo.getTitle());  put(**"reserveDate"**, reserveDate);  put(**"name"**, orderInfo.getPatientName());  }};  msmVo.setParam(param);  **rabbitService**.sendMessage(MqConst.***EXCHANGE\_DIRECT\_MSM***, MqConst.***ROUTING\_MSM\_ITEM***, msmVo);  }  } |

### 2.3 添加mq监听

添加OrderReceiver 类

|  |
| --- |
| @Component  **public class** OrderReceiver {  @Autowired  **private** OrderService **orderService**;  @RabbitListener(bindings = @QueueBinding(  value = @Queue(value = MqConst.***QUEUE\_TASK\_8***, durable = **"true"**),  exchange = @Exchange(value = MqConst.***EXCHANGE\_DIRECT\_TASK***),  key = {MqConst.***ROUTING\_TASK\_8***}  ))  **public void** patientTips(Message message, Channel channel) **throws** IOException {  **orderService**.patientTips();  }  } |

# 预约统计

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，因为在实际的生成环境中，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、ECharts

### 1.1简介

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](https://echarts.baidu.com/option.html" \l "series-line" \t "_blank)、[柱状图](https://echarts.baidu.com/option.html" \l "series-line" \t "_blank)、[散点图](https://echarts.baidu.com/option.html" \l "series-scatter" \t "_blank)、[饼图](https://echarts.baidu.com/option.html" \l "series-pie" \t "_blank)、[K线图](https://echarts.baidu.com/option.html" \l "series-candlestick" \t "_blank)，用于统计的[盒形图](https://echarts.baidu.com/option.html" \l "series-boxplot" \t "_blank)，用于地理数据可视化的[地图](https://echarts.baidu.com/option.html" \l "series-map" \t "_blank)、[热力图](https://echarts.baidu.com/option.html" \l "series-heatmap" \t "_blank)、[线图](https://echarts.baidu.com/option.html" \l "series-lines" \t "_blank)，用于关系数据可视化的[关系图](https://echarts.baidu.com/option.html" \l "series-graph" \t "_blank)、[treemap](https://echarts.baidu.com/option.html" \l "series-treemap" \t "_blank)、[旭日图](https://echarts.baidu.com/option.html" \l "series-sunburst)，多维数据可视化的[平行坐标](https://echarts.baidu.com/option.html" \l "series-parallel" \t "_blank)，还有用于 BI 的[漏斗图](https://echarts.baidu.com/option.html" \l "series-funnel" \t "_blank)，[仪表盘](https://echarts.baidu.com/option.html" \l "series-gauge" \t "_blank)，并且支持图与图之间的混搭。

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

### 1.2基本使用

（1）引入ECharts

|  |
| --- |
| <!-- 引入 ECharts 文件 -->  <script src="echarts.min.js"></script> |

（2）定义图表区域

|  |
| --- |
| <!-- 为ECharts准备一个具备大小（宽高）的Dom -->  <div id="main" style="width: 600px;height:400px;"></div> |

（3）渲染图表（折线图）

|  |
| --- |
| <script>      var myChart = echarts.init(document.getElementById('main'));      var option = {          //x轴是类目轴（离散数据）,必须通过data设置类目数据          xAxis: {              type: 'category',              data: ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun']          },          //y轴是数据轴（连续数据）          yAxis: {              type: 'value'          },          //系列列表。每个系列通过 type 决定自己的图表类型          series: [{              //系列中的数据内容数组              data: [820, 932, 901, 934, 1290, 1330, 1320],              //折线图              type: 'line'          }]      };      myChart.setOption(option);  </script> |

（4）渲染图表（柱状图）

|  |
| --- |
| <script type="text/javascript">      // 基于准备好的dom，初始化echarts实例      var myChart = echarts.init(document.getElementById('main'));      // 指定图表的配置项和数据      var option = {          title: {              text: 'ECharts 入门示例'          },          tooltip: {},          legend: {              data:['销量']          },          xAxis: {              data: ["衬衫","羊毛衫","雪纺衫","裤子","高跟鞋","袜子"]          },          yAxis: {},          series: [{              name: '销量',              type: 'bar',              data: [5, 20, 36, 10, 10, 20]          }]      };      // 使用刚指定的配置项和数据显示图表。      myChart.setOption(option);  </script> |

### 1.3项目中集成ECharts

|  |
| --- |
| npm install --save echarts@4.1.0 |

## 2、获取医院每天平台预约数据接口

操作模块：service-order

### 2.1添加Mapper接口

1、在OrderInfoMapper类添加接口

|  |
| --- |
| **public interface** OrderMapper **extends** BaseMapper<OrderInfo> {  List<OrderCountVo> selectOrderCount(@Param(**"vo"**) OrderCountQueryVo orderCountQueryVo);  } |

1. 在OrderInfoMapper.xml文件添加方法

|  |
| --- |
| *<?***xml version="1.0" encoding="UTF-8"** *?>*  **<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"**  **"http://mybatis.org/dtd/mybatis-3-mapper.dtd"*>***  <**mapper namespace="com.atguigu.yygh.order.mapper.OrderMapper"**>  <**select id="selectOrderCount" resultType="com.atguigu.yygh.vo.order.OrderCountVo"**>  select reserve\_date as reserveDate, *count*(reserve\_date) as count  from order\_info  <**where**>  <**if test="vo.hosname != null and vo.hosname != ''"**>  and hosname like CONCAT('%',#{vo.hosname},'%')  </**if**>  <**if test="vo.reserveDateBegin != null and vo.reserveDateBegin != ''"**>  and reserve\_date >= #{vo.reserveDateBegin}  </**if**>  <**if test="vo.reserveDateEnd != null and vo.reserveDateEnd != ''"**>  and reserve\_date **&lt;**= #{vo.reserveDateEnd}  </**if**>  and is\_deleted = 0  </**where**>  group by reserve\_date  order by reserve\_date  </**select**>  </**mapper**> |

添加application.properties配置

|  |
| --- |
| **mybatis-plus.mapper-locations**=**classpath:com/atguigu/yygh/order/mapper/xml/\*.xml** |

### 2.2添加service接口

在OrderService类添加接口

|  |
| --- |
| */\*\**  *\* 订单统计*  *\*/*  Map<String, Object> getCountMap(OrderCountQueryVo orderCountQueryVo); |

### 2.3添加service接口实现

在OrderServiceImpl类添加实现

|  |
| --- |
| @Override  **public** Map<String, Object> getCountMap(OrderCountQueryVo orderCountQueryVo) {  Map<String, Object> map = **new** HashMap<>();  List<OrderCountVo> orderCountVoList  = **baseMapper**.selectOrderCount(orderCountQueryVo);  *//日期列表*  List<String> dateList  =orderCountVoList.stream().map(OrderCountVo::getReserveDate).collect(Collectors.*toList*());  *//统计列表*  List<Integer> countList  =orderCountVoList.stream().map(OrderCountVo::getCount).collect(Collectors.*toList*());  map.put(**"dateList"**, dateList);  map.put(**"countList"**, countList);  **return** map;  } |

### 2.4添加controller方法

在OrderApiController类添加方法

|  |
| --- |
| @ApiOperation(value = **"获取订单统计数据"**)  @PostMapping(**"inner/getCountMap"**)  **public** Map<String, Object> getCountMap(@RequestBody OrderCountQueryVo orderCountQueryVo) {  **return orderService**.getCountMap(orderCountQueryVo);  } |

## 3、添加feign方法

创建模块：service-order-client

### 3.1添加feign接口

添加接口和方法

|  |
| --- |
| @FeignClient(value = **"service-order"**)  @Repository  **public interface** OrderFeignClient {  */\*\**  *\* 获取订单统计数据*  *\*/*  @PostMapping(**"/api/order/orderInfo/inner/getCountMap"**)  Map<String, Object> getCountMap(@RequestBody OrderCountQueryVo orderCountQueryVo);  } |

## 4、搭建service-statistics

### 4.1搭建service-statistics服务

搭建方式如service-user

### 4.2修改配置pom.xml

|  |
| --- |
| <**dependencies**>  <**dependency**>  <**groupId**>com.atguigu</**groupId**>  <**artifactId**>service\_order\_client</**artifactId**>  <**version**>0.0.1-SNAPSHOT</**version**>  </**dependency**>  </**dependencies**> |

### 4.3添加配置文件

1、application.properties

|  |
| --- |
| *# 服务端口*  **server.port**=**8208**  *# 服务名*  **spring.application.name**=**service-statistics**  *# 环境设置：dev、test、prod*  **spring.profiles.active**=**dev**  *# nacos服务地址*  **spring.cloud.nacos.discovery.server-addr**=**127.0.0.1:8848** |

### 4.4添加启动类

|  |
| --- |
| @SpringBootApplication(exclude = DataSourceAutoConfiguration.**class**)*//取消数据源自动配置*  @EnableDiscoveryClient  @EnableFeignClients  @ComponentScan(basePackages = {**"com.atguigu"**})  **public class** ServiceStatisticsApplication {  **public static void** main(String[] args) {  SpringApplication.*run*(ServiceStatisticsApplication.**class**, args);  }  } |

### 4.5添加controller方法

|  |
| --- |
| @Api(tags = **"统计管理接口"**)  @RestController  @RequestMapping(**"/admin/statistics"**)  **public class** StatisticsController {  @Autowired  **private** OrderFeignClient **orderFeignClient**;  @ApiOperation(value = **"获取订单统计数据"**)  @GetMapping(**"getCountMap"**)  **public** Result getCountMap(@ApiParam(name = **"orderCountQueryVo"**, value = **"查询对象"**, required = **false**) OrderCountQueryVo orderCountQueryVo) {  **return** Result.*ok*(**orderFeignClient**.getCountMap(orderCountQueryVo));  }  } |

## 5、前端展示

### 5.1添加路由

在 src/router/index.js 文件添加路由

|  |
| --- |
| {    path: '/statistics',    component: Layout,    redirect: '/statistics/order/index',    name: 'BasesInfo',    meta: { title: '统计管理', icon: 'table' },    alwaysShow: true,    children: [        {    path: 'order/index',    name: '预约统计',    component: () =>import('@/views/statistics/order/index'),    meta: { title: '预约统计' }        }      ]  }, |

### 5.2封装api请求

创建/api/statistics/orderStatistics.js

|  |
| --- |
| import request from '@/utils/request'  const api\_name = '/admin/statistics'  export default {      getCountMap(searchObj) {          return request({              url: `${api\_name}/getCountMap`,              method: 'get',              params: searchObj          })      }  } |

### 5.3添加组件

创建/views/statistics/order/index.vue组件

|  |
| --- |
| <template>      <div class="app-container">      <!--表单-->      <el-form :inline="true" class="demo-form-inline">          <el-form-item>              <el-input v-model="searchObj.hosname" placeholder="点击输入医院名称"/>          </el-form-item>          <el-form-item>              <el-date-picker                  v-model="searchObj.reserveDateBegin"                  type="date"                  placeholder="选择开始日期"                  value-format="yyyy-MM-dd"/>          </el-form-item>          <el-form-item>              <el-date-picker                  v-model="searchObj.reserveDateEnd"                  type="date"                  placeholder="选择截止日期"                  value-format="yyyy-MM-dd"/>          </el-form-item>          <el-button              :disabled="btnDisabled"              type="primary"              icon="el-icon-search"              @click="showChart()">查询</el-button>      </el-form>      <div class="chart-container">          <div id="chart" ref="chart"              class="chart" style="height:500px;width:100%"/>      </div>      </div>  </template>  <script>  import echarts from 'echarts'  import statisticsApi from '@/api/orderStatistics'  export default {      data() {          return {              searchObj: {                  hosname: '',                  reserveDateBegin: '',                  reserveDateEnd: ''              },              btnDisabled: false,              chart: null,              title: '',              xData: [], // x轴数据              yData: [] // y轴数据          }      },      methods: {          // 初始化图表数据          showChart() {              statisticsApi.getCountMap(this.searchObj).then(response => {                  this.yData = response.data.countList                  this.xData = response.data.dateList                  this.setChartData()              })          },          setChartData() {              // 基于准备好的dom，初始化echarts实例              var myChart = echarts.init(document.getElementById('chart'))              // 指定图表的配置项和数据              var option = {                  title: {                      text: this.title + '挂号量统计'                  },                  tooltip: {},                  legend: {                      data: [this.title]                  },                  xAxis: {                      data: this.xData                  },                  yAxis: {                      minInterval: 1                  },                  series: [{                      name: this.title,                      type: 'line',                      data: this.yData                  }]              }              // 使用刚指定的配置项和数据显示图表。              myChart.setOption(option)          },      }  }  </script> |