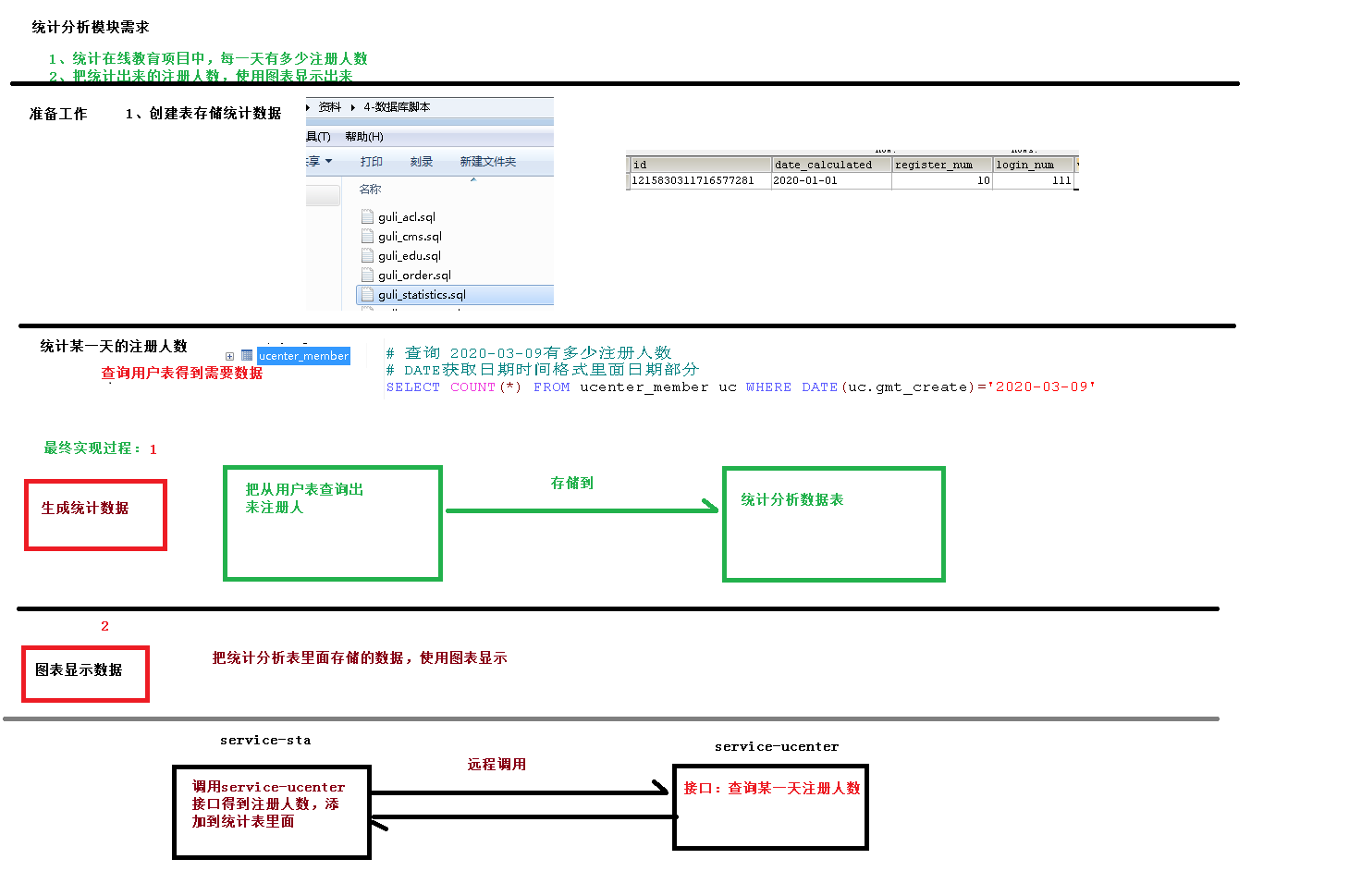
day16项目【统计分析】



# 01.统计分析功能（生成统计数据）

## 一、数据库设计

### 1、数据库

guli\_statistics

### 2、数据表

guli\_statistics.sql

CREATE TABLE `statistics\_daily` (

  `id` char(19) NOT NULL COMMENT '主键',

  `date\_calculated` varchar(20) NOT NULL COMMENT '统计日期',

  `register\_num` int(11) NOT NULL DEFAULT '0' COMMENT '注册人数',

  `login\_num` int(11) NOT NULL DEFAULT '0' COMMENT '登录人数',

  `video\_view\_num` int(11) NOT NULL DEFAULT '0' COMMENT '每日播放视频数',

  `course\_num` int(11) NOT NULL DEFAULT '0' COMMENT '每日新增课程数',

  `gmt\_create` datetime NOT NULL COMMENT '创建时间',

  `gmt\_modified` datetime NOT NULL COMMENT '更新时间',

  PRIMARY KEY (`id`),

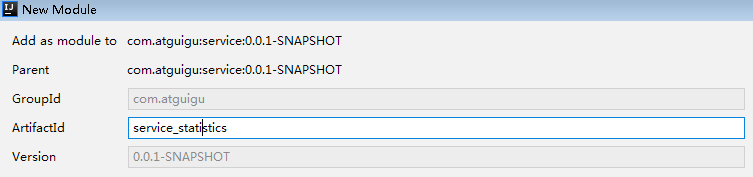
  KEY `statistics\_day` (`date\_calculated`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='网站统计日数据';

## 二、创建微服务

### 1、在service模块下创建子模块

service\_statistics



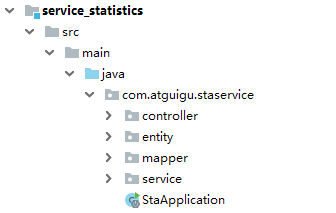
### 2、application.properties

resources目录下创建文件

# 服务端口

server.port=8008  
# 服务名  
spring.application.name=service-statistics  
# mysql数据库连接  
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver  
spring.datasource.url=jdbc:mysql://localhost:3306/guli?serverTimezone=GMT%2B8  
spring.datasource.username=root  
spring.datasource.password=root  
#返回json的全局时间格式  
spring.jackson.date-format=yyyy-MM-dd HH:mm:ss  
spring.jackson.time-zone=GMT+8  
#配置mapper xml文件的路径  
mybatis-plus.mapper-locations=classpath:com/atguigu/staservice/mapper/xml/\*.xml  
#mybatis日志  
mybatis-plus.configuration.log-impl=org.apache.ibatis.logging.stdout.StdOutImpl  
# nacos服务地址  
spring.cloud.nacos.discovery.server-addr=127.0.0.1:8848  
#开启熔断机制  
feign.hystrix.enabled=true  
# 设置hystrix超时时间，默认1000ms  
hystrix.command.default.execution.isolation.thread.timeoutInMilliseconds=3000

### 3、MP代码生成器生成代码



### 4、创建SpringBoot启动类

@SpringBootApplication

@MapperScan*(*"com.atguigu.staservice.mapper"*)*@ComponentScan*(*"com.atguigu"*)*@EnableDiscoveryClient  
@EnableFeignClients  
public class StaApplication *{* public static void main*(*String*[]* args*) {* SpringApplication.*run(*StaApplication.class, args*)*;  
 *}  
}*

## 三、实现服务调用

### 1、在service\_ucenter模块创建接口，统计某一天的注册人数

#### controller

@GetMapping(value = "countregister/{day}")

public R registerCount(  
 @PathVariable String day) {  
 Integer count = memberService.countRegisterByDay(day);  
 return R.ok().data("countRegister", count);  
}

#### service

@Override  
public Integer countRegisterByDay*(*String day*) {* return baseMapper.selectRegisterCount*(*day*)*;  
*}*

#### mapper

*<*select id="selectRegisterCount" resultType="java.lang.Integer"*>* SELECT COUNT(1)  
 FROM ucenter\_member  
 WHERE DATE(gmt\_create) = #{value}  
*</*select*>*

### 2、在service\_statistics模块创建远程调用接口

创建client包和UcenterClient接口

@Component

@FeignClient("service-ucenter")  
public interface UcenterClient {  
 @GetMapping(value = "/ucenterservice/member/countregister/{day}")  
 public R registerCount(@PathVariable("day") String day);  
}

### 3、在service\_statistics模块调用微服务

#### controller

@PostMapping("{day}")

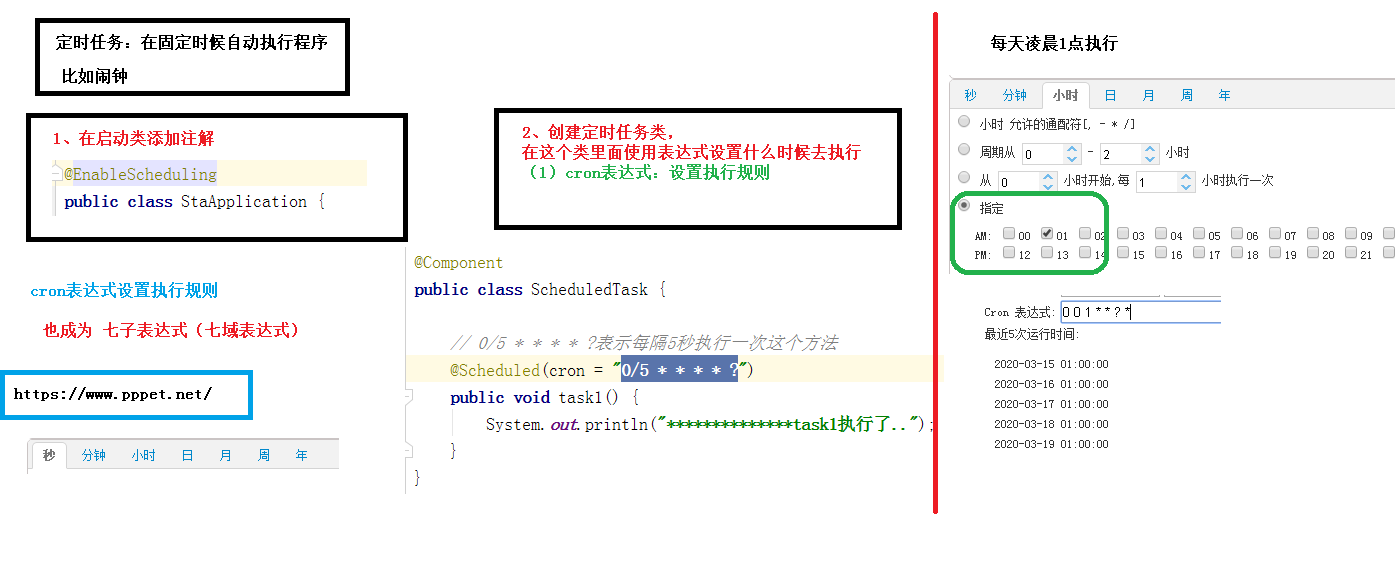
public R createStatisticsByDate(@PathVariable String day) {  
 dailyService.createStatisticsByDay(day);  
 return R.ok();  
}

#### service

@Service

public class StatisticsDailyServiceImpl extends ServiceImpl*<*StatisticsDailyMapper, StatisticsDaily*>* implements StatisticsDailyService *{* @Autowired  
 private UcenterClient ucenterClient;  
  
 @Override  
 public void createStatisticsByDay*(*String day*) {* //删除已存在的统计对象  
 QueryWrapper*<*StatisticsDaily*>* dayQueryWrapper = new QueryWrapper*<>()*;  
 dayQueryWrapper.eq*(*"date\_calculated", day*)*;  
 baseMapper.delete*(*dayQueryWrapper*)*;  
 //获取统计信息  
 Integer registerNum = *(*Integer*)* ucenterClient.registerCount*(*day*)*.getData*()*.get*(*"countRegister"*)*;  
 Integer loginNum = RandomUtils.*nextInt(*100, 200*)*;//*TODO* Integer videoViewNum = RandomUtils.*nextInt(*100, 200*)*;//*TODO* Integer courseNum = RandomUtils.*nextInt(*100, 200*)*;//*TODO* //创建统计对象  
 StatisticsDaily daily = new StatisticsDaily*()*;  
 daily.setRegisterNum*(*registerNum*)*;  
 daily.setLoginNum*(*loginNum*)*;  
 daily.setVideoViewNum*(*videoViewNum*)*;  
 daily.setCourseNum*(*courseNum*)*;  
 daily.setDateCalculated*(*day*)*;  
 baseMapper.insert*(*daily*)*;  
 *}  
}*

## 四、添加定时任务



### 1、创建定时任务类，使用cron表达式

**复制日期工具类**

@Component

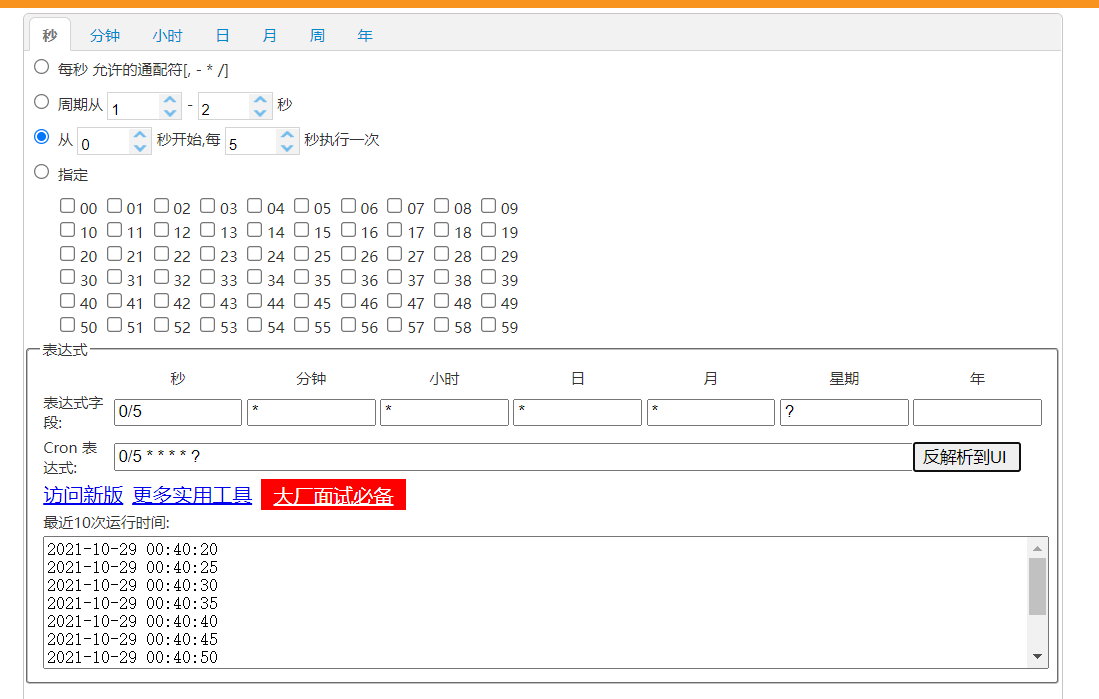
public class ScheduledTask *{* @Autowired  
 private StatisticsDailyService dailyService;  
  
 */\*\*  
 \* 测试  
 \*每五秒执行一次  
 \*/* @Scheduled*(*cron = "0/5 \* \* \* \* ?"*)//6位 默认当前年* public void task1*() {* System.*out*.println*(*"\*\*\*\*\*\*\*\*\*++++++++++++\*\*\*\*\*执行了"*)*;  
 *}  
  
 /\*\*  
 \* 每天凌晨1点执行定时  
 \*/* @Scheduled*(*cron = "0 0 1 \* \* ?"*)* public void task2*() {* //获取上一天的日期  
 String day = DateUtil.formatDate*(*DateUtil.addDays*(*new Date*()*, -1*))*;  
 dailyService.createStatisticsByDay*(*day*)*;  
 *}  
}*

### 2、在启动类上添加注解@EnableScheduling

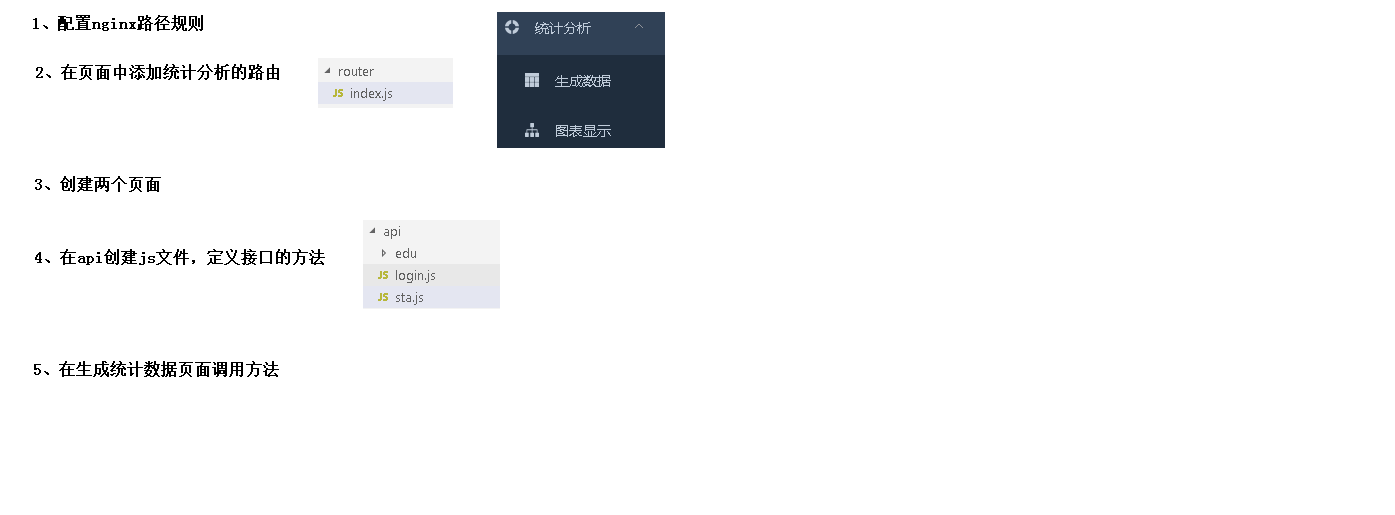
C:\Users\gouwenyong0000\Documents\My Knowledge\temp\68cdac17-2239-4330-af25-83862ccc0ebf\128\index_files\bc8246ec-d993-43f6-a9dd-61f81b622c12.png

### 3、在线生成cron表达式

http://cron.qqe2.com/



# 02.生成统计数据前端整合



## 一、nginx配置

location ~ /staservice/ {

proxy\_pass http://localhost:8008;

}

## 二、前端页面实现

### 1、创建api

创建src/api/sta.js

import request from '@/utils/request'

*const* api\_name = '/admin/statistics/daily'

export default {

  createStatistics(*day*) {

    return request({

      url: `${api\_name}/${*day*}`,

      method: 'post'

    })

  }

}

### 2、增加路由

src/router/index.js

{

     path: '/statistics/daily',

     component: Layout,

     redirect: '/statistics/daily/create',

     name: 'Statistics',

     meta: { title: '统计分析', icon: 'chart' },

     children: [

       {

         path: 'create',

         name: 'StatisticsDailyCreate',

         component: () *=>* import('@/views/statistics/daily/create'),

         meta: { title: '生成统计' }

       }

     ]

   },

### 3、创建组件

src/views/statistics/daily/create.vue

模板部分

<template>

<div class="app-container">

<!--表单-->

<el-form :inline="true" class="demo-form-inline">

<el-form-item label="日期">

<el-date-picker

v-model="day"

type="date"

placeholder="选择要统计的日期"

value-format="yyyy-MM-dd" />

</el-form-item>

<el-button

:disabled="btnDisabled"

type="primary"

@click="create()">生成</el-button>

</el-form>

</div>

</template>

script部分

<script>

import daily from "@/api/edu/sta.js";

export default {

  data() {

    return {

      day: '',

      btnDisabled: false

    }

  },

  methods: {

    create() {

      this.btnDisabled = true

      daily.createStatistics(this.day).then(*response* *=>* {

        this.btnDisabled = false

        this.$message({

          type: 'success',

          message: '生成成功'

        })

      })

    }

  }

}

</script>

# 03.统计数据图表显示

## 一、ECharts

### 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#series-line)、[柱状图](https://echarts.baidu.com/option.html#series-line)、[散点图](https://echarts.baidu.com/option.html#series-scatter)、[饼图](https://echarts.baidu.com/option.html#series-pie)、[K线图](https://echarts.baidu.com/option.html#series-candlestick)，用于统计的[盒形图](https://echarts.baidu.com/option.html#series-boxplot)，用于地理数据可视化的[地图](https://echarts.baidu.com/option.html#series-map)、[热力图](https://echarts.baidu.com/option.html#series-heatmap)、[线图](https://echarts.baidu.com/option.html#series-lines)，用于关系数据可视化的[关系图](https://echarts.baidu.com/option.html#series-graph)、[treemap](https://echarts.baidu.com/option.html#series-treemap)、[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#series-parallel)，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#series-gauge)，并且支持图与图之间的混搭。

官方网站：<https://echarts.baidu.com/>

### 2、基本使用

入门参考：官网->文档->教程->5分钟上手ECharts

（1）创建html页面：柱图.html

（2）引入ECharts

<!-- 引入 ECharts 文件 -->

<script src="echarts.min.js"></script>

（3）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>

（4）渲染图表

<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>

### 3、折线图

实例参考：官网->实例->官方实例 <https://echarts.baidu.com/examples/>

折线图.html

<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'

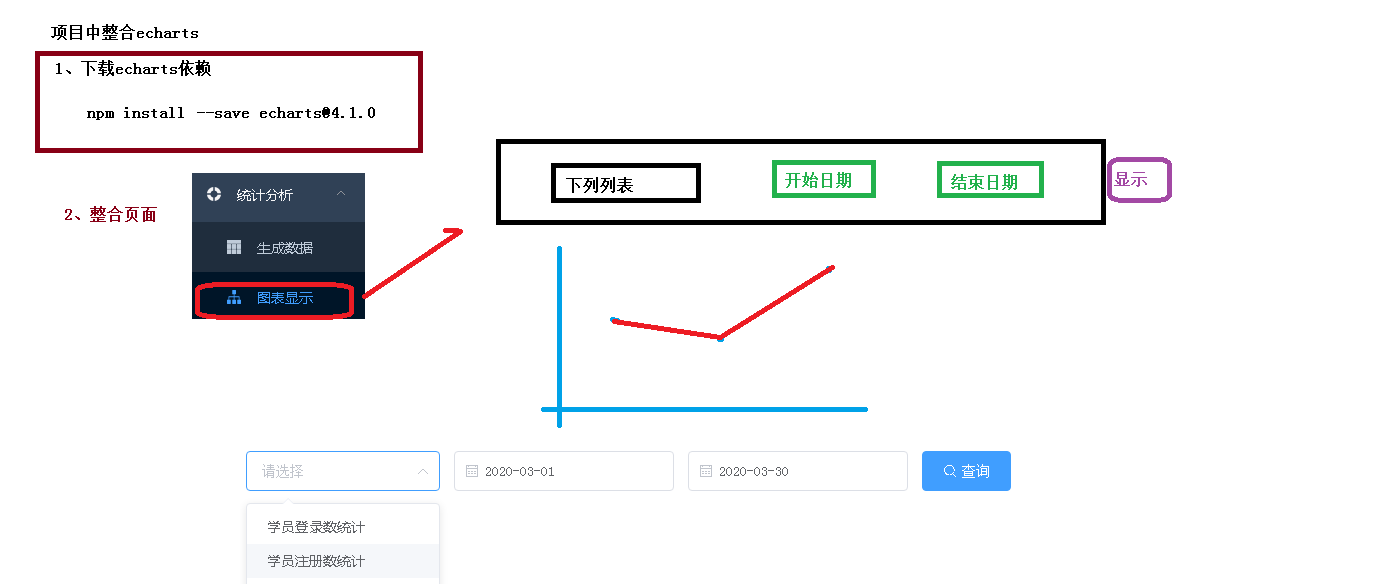
        }]

    };

    myChart.setOption(option);

</script>

## 二、项目中集成ECharts



### 1、安装ECharts

npm install --save echarts@4.1.0

### 2、增加路由

src/router/index.js

在统计分析路由中增加子路由

{

path: 'chart',

name: 'StatisticsDayChart',

component: () => import('@/views/statistics/daily/chart'),

meta: { title: '统计图表' }

}

### 3、创建组件

src/views/statistics/daily/chart.vue

**模板**

<template>

  <div class="app-container">

    <!--表单-->

    <el-form :inline="true" class="demo-form-inline">

      <el-form-item>

        <el-select v-model="searchObj.type" clearable placeholder="请选择">

          <el-option label="学员登录数统计" value="login\_num"/>

          <el-option label="学员注册数统计" value="register\_num"/>

          <el-option label="课程播放数统计" value="video\_view\_num"/>

          <el-option label="每日课程数统计" value="course\_num"/>

        </el-select>

      </el-form-item>

      <el-form-item>

        <el-date-picker

          v-model="searchObj.begin"

          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.end"

          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" class="chart" style="height:500px;width:100%" />

    </div>

  </div>

</template>

**js：暂时显示临时数据**

<script>

import echarts from 'echarts'

export default {

  data() {

    return {

      searchObj: {

        type: '',

        begin: '',

        end: ''

      },

      btnDisabled: false,

      chart: null,

      title: '',

      xData: [],

      yData: []

    }

  },

  methods: {

    showChart() {

      this.initChartData()

      this.setChart()

    },

    // 准备图表数据

    initChartData() {

    },

    // 设置图标参数

    setChart() {

      // 基于准备好的dom，初始化echarts实例

      this.chart = echarts.init(document.getElementById('chart'))

      // console.log(this.chart)

      // 指定图表的配置项和数据

*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'

        }]

      }

      this.chart.setOption(option)

    }

  }

}

</script>

## 三、完成后端业务

### 1、controller

@GetMapping("show-chart/{begin}/{end}/{type}")

public R showChart(@PathVariable String begin, @PathVariable String end, @PathVariable String type) {  
 Map<String, Object> map = dailyService.getChartData(begin, end, type);  
 return R.ok().data(map);  
}

### 2、service

**接口**

Map<String, Object> getChartData(String begin, String end, String type);

**实现**

@Override

public Map*<*String, Object*>* getChartData*(*String begin, String end, String type*) {* QueryWrapper*<*Daily*>* dayQueryWrapper = new QueryWrapper*<*>*()*;  
 dayQueryWrapper.select*(*type, "date\_calculated"*)*;  
 dayQueryWrapper.between*(*"date\_calculated", begin, end*)*;  
 List*<*Daily*>* dayList = baseMapper.selectList*(*dayQueryWrapper);  
 Map*<*String, Object*>* map = new HashMap*<>()*;  
 List*<*Integer*>* dataList = new ArrayList*<*Integer*>()*;  
 List*<*String*>* dateList = new ArrayList*<*String*>()*;  
 map.put*(*"dataList", dataList*)*;  
 map.put*(*"dateList", dateList*)*;  
 for *(*int i = 0; i < dayList.size*()*; i++*) {* Daily daily = dayList.get*(*i*)*;  
 dateList.add*(*daily.getDateCalculated*())*;  
 switch *(*type*) {* case "register\_num":  
 dataList.add*(*daily.getRegisterNum*())*;  
 break;  
 case "login\_num":  
 dataList.add*(*daily.getLoginNum*())*;  
 break;  
 case "video\_view\_num":  
 dataList.add*(*daily.getVideoViewNum*())*;  
 break;  
 case "course\_num":  
 dataList.add*(*daily.getCourseNum*())*;  
 break;  
 default:  
 break;  
 *}  
 }* return map;  
*}*

## 四、前后端整合

### 1、创建api

src/api/statistics/daily.js中添加方法

showChart(*searchObj*) {

    return request({

        url: `${api\_name}/show-chart/${*searchObj*.begin}/${*searchObj*.end}/${*searchObj*.type}`,

        method: 'get'

    })

}

### 2、chart.vue中引入api模块

import daily from '@/api/statistics/daily'

......

### 3、修改initChartData方法

showChart() {

  this.initChartData()

  // this.setChart()//放在initChartData回调中执行

},

// 准备图表数据

initChartData() {

  daily.showChart(this.searchObj).then(*response* *=>* {

    // 数据

    this.yData = *response*.data.dataList

    // 横轴时间

    this.xData = *response*.data.dateList

    // 当前统计类别

    switch (this.searchObj.type) {

      case 'register\_num':

        this.title = '学员注册数统计'

        break

      case 'login\_num':

        this.title = '学员登录数统计'

        break

      case 'video\_view\_num':

        this.title = '课程播放数统计'

        break

      case 'course\_num':

        this.title = '每日课程数统计'

        break

    }

    this.setChart()

  })

},

### 4、修改options中的数据

xAxis: {

    type: 'category',

    data: this.xData//-------绑定数据

},

// 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: this.yData,//-------绑定数据

    // 折线图

    type: 'line'

}],

## 五、样式调整

参考配置手册：[https://echarts.baidu.com/option.html#title](https://echarts.baidu.com/option.html" \l "title)

### 1、显示标题

        title: {

          text: this.title,

        },

### 2、x坐标轴触发提示

tooltip: {

    trigger: 'axis'

},

### 3、区域缩放

        dataZoom: [

          {

            show: true,

            height: 30,

            xAxisIndex: [0],

            bottom: 30,

            start: 10,

            end: 80,

            handleIcon:

              "path://M306.1,413c0,2.2-1.8,4-4,4h-59.8c-2.2,0-4-1.8-4-4V200.8c0-2.2,1.8-4,4-4h59.8c2.2,0,4,1.8,4,4V413z",

            handleSize: "110%",

            handleStyle: {

              color: "#d3dee5",

            },

            textStyle: {

              color: "#fff",

            },

            borderColor: "#90979c",

          },

          {

            type: "inside",

            show: true,

            height: 15,

            start: 1,

            end: 35,

          },

        ],