**实验报告三**

学号：2019329621004 姓名：梅雨欣

班级：19计算机科学与技术（1） 联系电话：13007239653

日期：2022.04.01 上课节次：周五12

**注意：**

1.上方的姓名、学号、班级、联系电话、日期、上课节次必须填写清晰准确；

2.文件名为“学号­\_姓名\_实验三.docx”，PDF格式也可以。不按照该规范命名，则视为未交回实验报告；

3.本次实验请在1周内完成，提交到蓝墨云班课。

1. **实验目标：**
2. 掌握使用Vue-CLI脚手架工具在自己的电脑上建立项目，并运行调试工具。
3. 学习使用ECharts图表开源库，绘制折线图或柱状图。数据自己选定。
4. 学习使用ECharts图表开源库开发地图展示功能，了解地图开发方法。基本用法参考ECharts官网：https://echarts.apache.org/zh/index.html
5. **实验内容：**

# 1.Vue-CLI脚手架工具搭建一个Web项目yiqing\_demo。

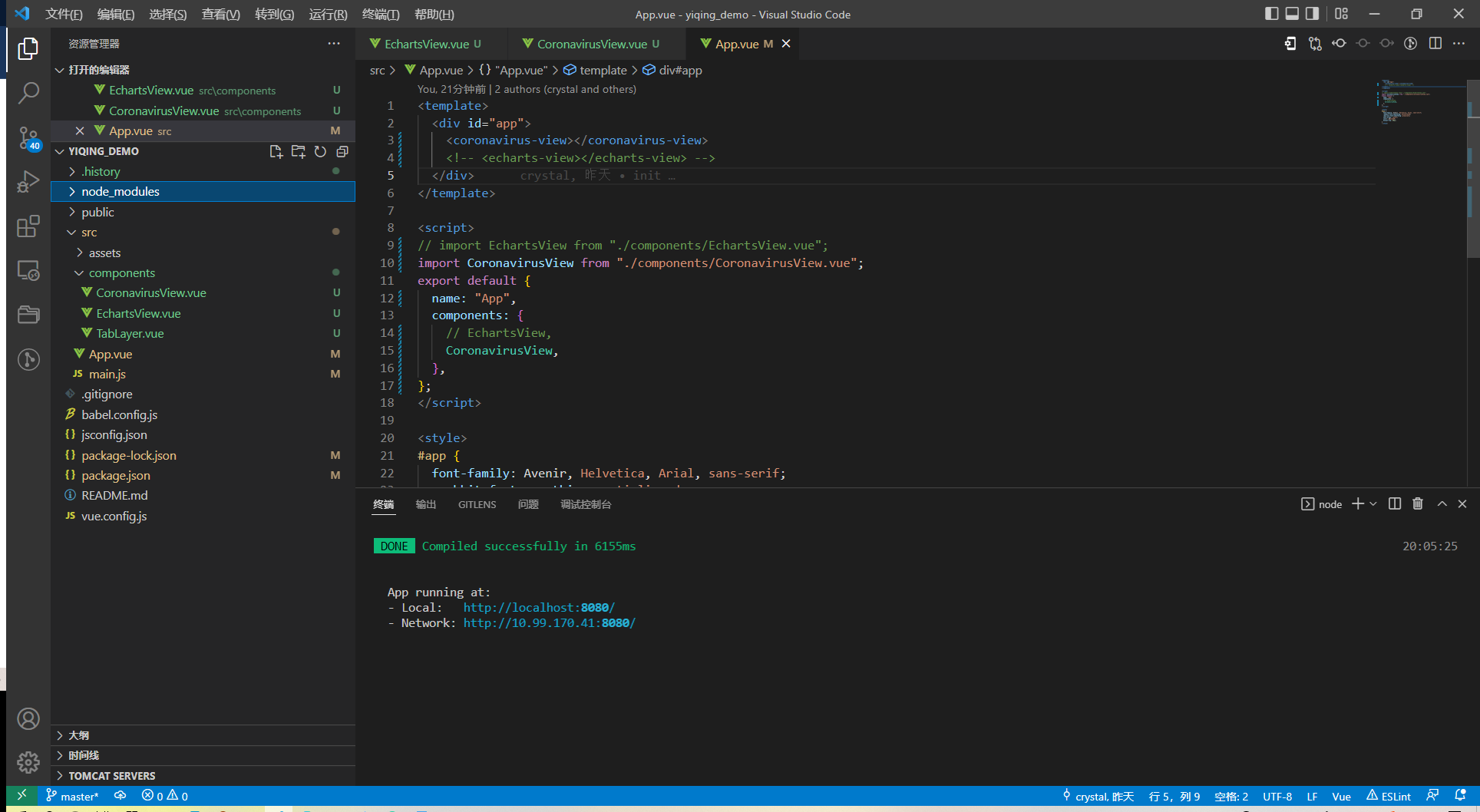
# 2.使用Vue.js和ECharts(地图工具推荐用ECharts4.x版本，也可以自己选定其他版本或其他工具)编写一个疫情地图显示网页。内容及格式可自己定，效果可参照丁香园疫情地图：

# https://ncov.dxy.cn/ncovh5/view/pneumonia

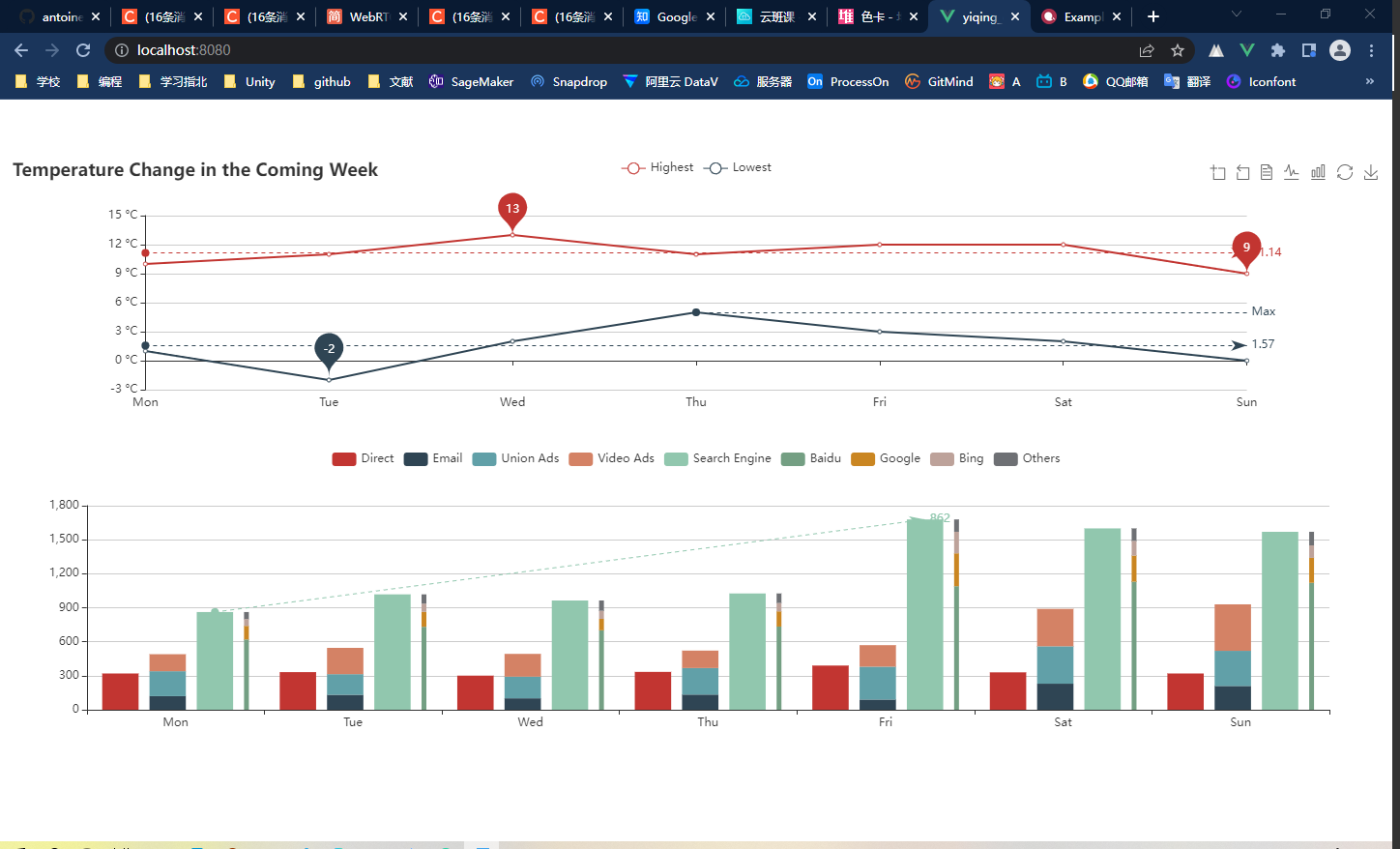
1. **实验过程：（描述详细步骤和成果截图）**

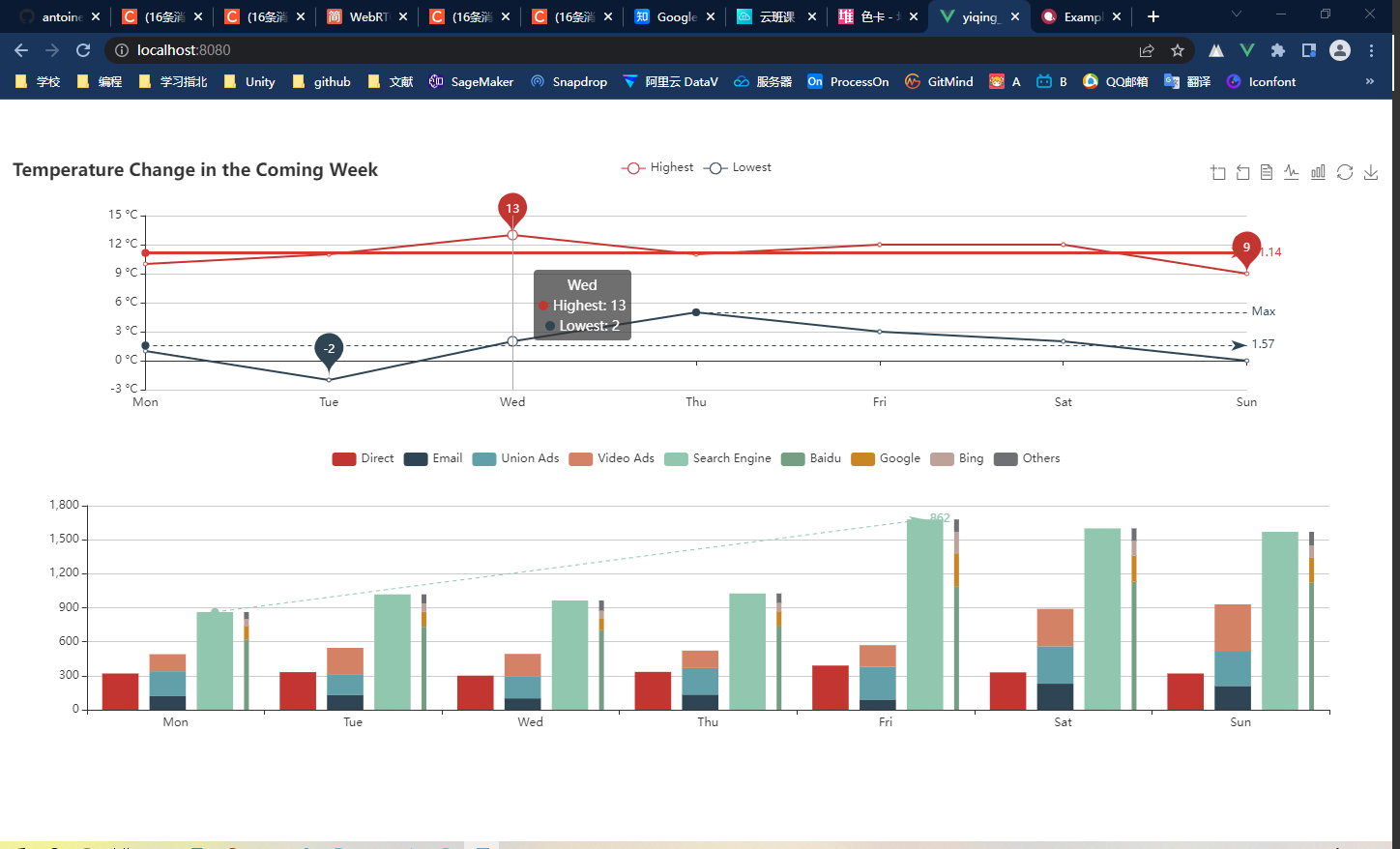
**1、Vue-CLI脚手架工具搭建Web项目yiqing\_demo**

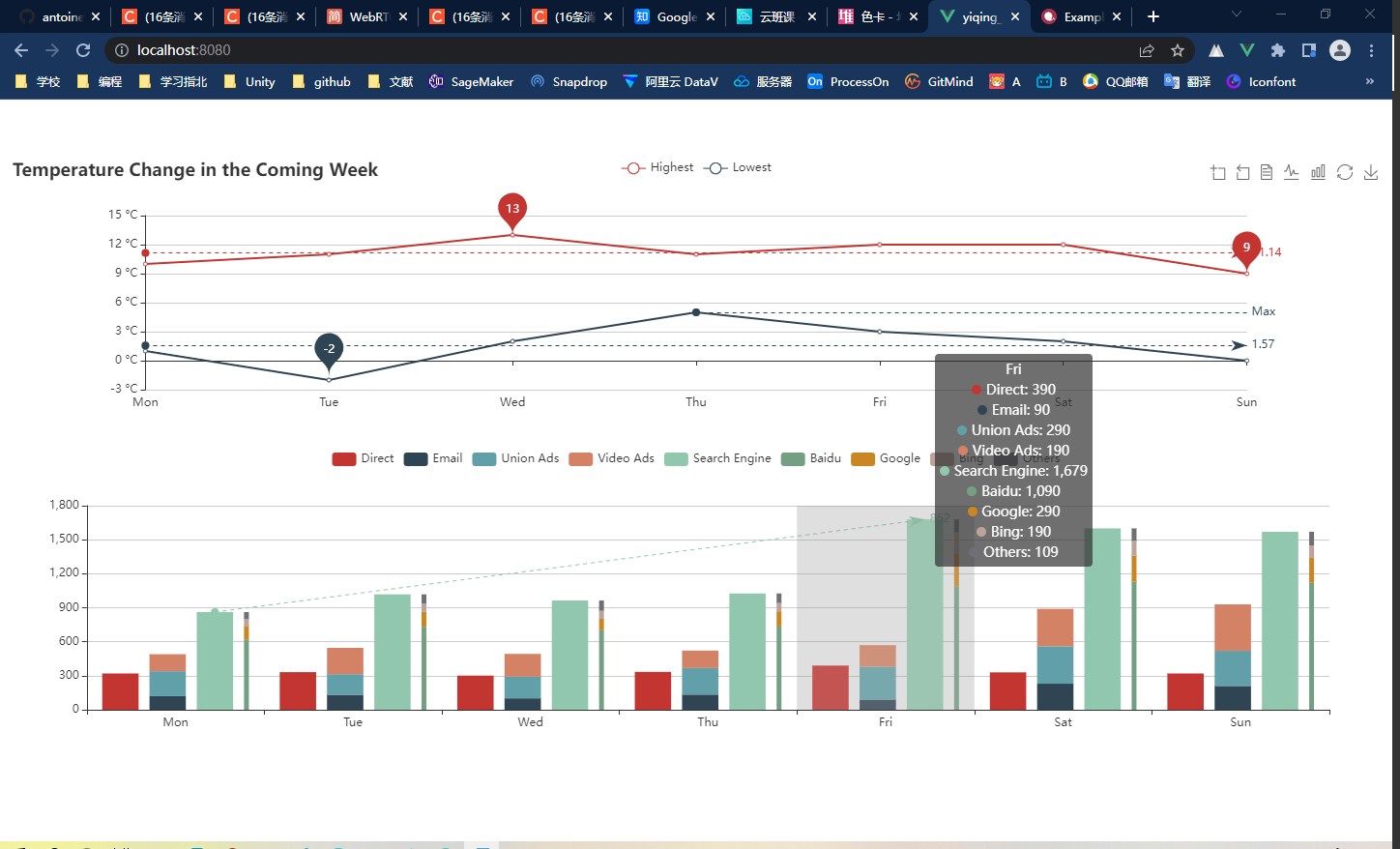
**输入vue create yiqing\_demo，创建成功。**



**2、使用ECharts绘制折线图和柱状图**

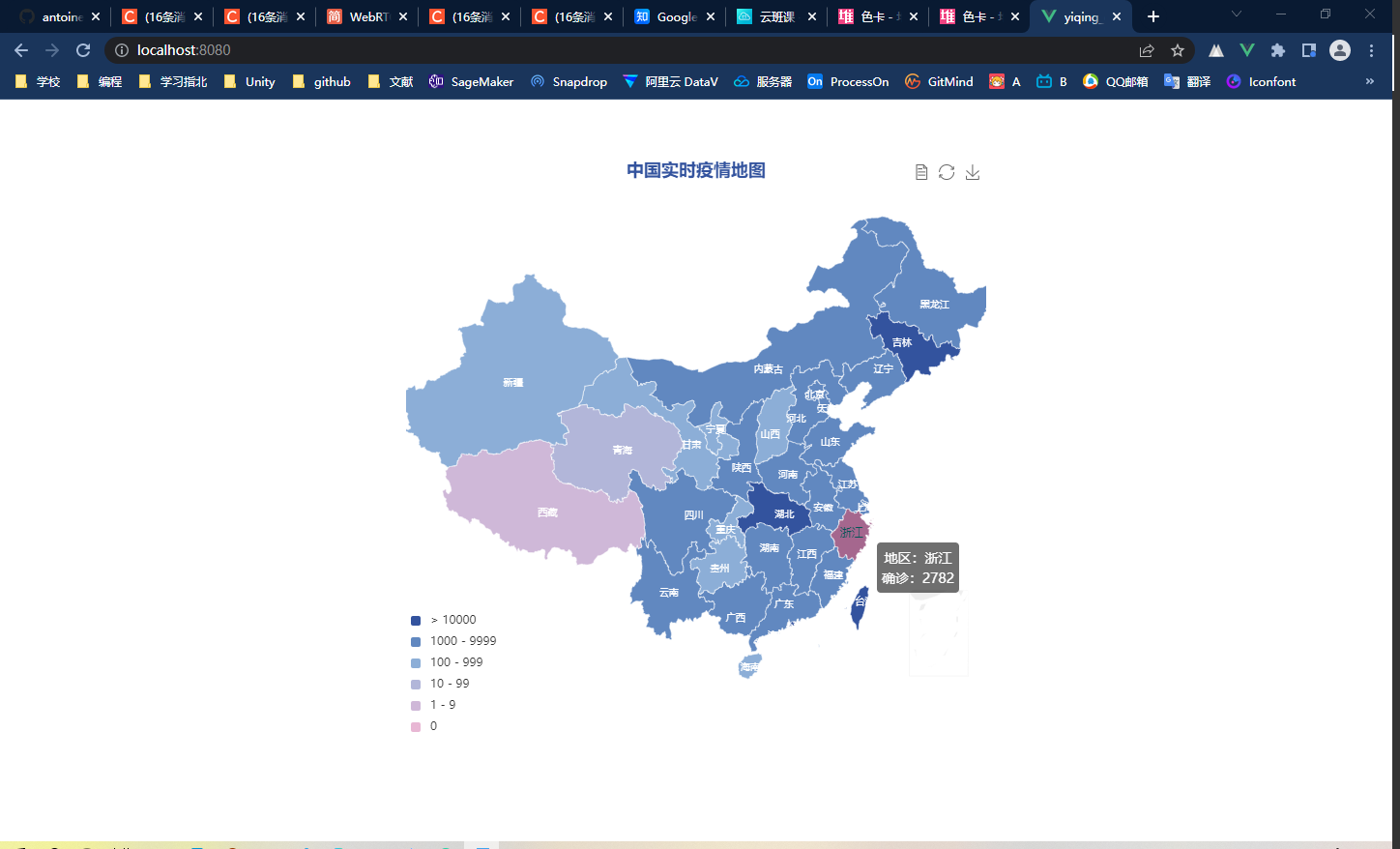






**3、使用ECharts制作中国疫情地图**





1. **心得体会：**

**通过这几次实验我已经能够熟练掌握vue-cli脚手架工具创建和调试Vue项目。我还学会了使用ECharts图表绘制可视化图表，如折线图、柱状图、热力图等，还能够开发疫情地图功能，熟悉了开发方法。ECharts图表美观清晰，易于上手直接操作。**

1. **代码：**

**App.vue**

|  |
| --- |
| <template>    <div id="app">      <!-- <coronavirus-view></coronavirus-view> -->      <echarts-view></echarts-view>    </div>  </template>  <script>  import EchartsView from "./components/EchartsView.vue";  // import CoronavirusView from "./components/CoronavirusView.vue";  export default {    name: "App",    components: {      EchartsView,      // CoronavirusView,    },  };  </script>  <style>  #app {    font-family: Avenir, Helvetica, Arial, sans-serif;    -webkit-font-smoothing: antialiased;    -moz-osx-font-smoothing: grayscale;    text-align: center;    color: #2c3e50;    margin-top: 60px;  }  </style> |

**EchartsView.vue**

|  |
| --- |
| <template>    <div class="about">      <div id="mychart1" style="width: 100%; height: 300px"></div>      <div id="mychart2" style="width: 100%; height: 300px"></div>    </div>  </template>  <script>  // @ is an alias to /src  // import TabLayer from '@/components/TabLayer.vue'  import \* as echarts from "echarts"; //引入echarts  export default {    name: "EchartsView",    data() {      return {};    },    mounted() {      //钩子函数，生命周期为挂载完成后，Dom操作一般是在mounted钩子函数中进行。      this.init();    },    methods: {      init() {        var myEchart1 = echarts.init(document.getElementById("mychart1"));        // 指定图表的配置项和数据        var option1 = {          title: {            text: 'Temperature Change in the Coming Week'          },          tooltip: {            trigger: 'axis'          },          legend: {},          toolbox: {            show: true,            feature: {              dataZoom: {                yAxisIndex: 'none'              },              dataView: { readOnly: false },              magicType: { type: ['line', 'bar'] },              restore: {},              saveAsImage: {}            }          },          xAxis: {            type: 'category',            boundaryGap: false,            data: ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun']          },          yAxis: {            type: 'value',            axisLabel: {              formatter: '{value} °C'            }          },          series: [            {              name: 'Highest',              type: 'line',              data: [10, 11, 13, 11, 12, 12, 9],              markPoint: {                data: [                  { type: 'max', name: 'Max' },                  { type: 'min', name: 'Min' }                ]              },              markLine: {                data: [{ type: 'average', name: 'Avg' }]              }            },            {              name: 'Lowest',              type: 'line',              data: [1, -2, 2, 5, 3, 2, 0],              markPoint: {                data: [{ name: '周最低', value: -2, xAxis: 1, yAxis: -1.5 }]              },              markLine: {                data: [                  { type: 'average', name: 'Avg' },                  [                    {                      symbol: 'none',                      x: '90%',                      yAxis: 'max'                    },                    {                      symbol: 'circle',                      label: {                        position: 'start',                        formatter: 'Max'                      },                      type: 'max',                      name: '最高点'                    }                  ]                ]              }            }          ]        };        // 使用刚指定的配置项和数据显示图表        myEchart1.setOption(option1);        var myEchart2 = echarts.init(document.getElementById("mychart2"));        // 指定图表的配置项和数据        var option2 = {          tooltip: {            trigger: 'axis',            axisPointer: {              type: 'shadow'            }          },          legend: {},          grid: {            left: '3%',            right: '4%',            bottom: '3%',            containLabel: true          },          xAxis: [            {              type: 'category',              data: ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun']            }          ],          yAxis: [            {              type: 'value'            }          ],          series: [            {              name: 'Direct',              type: 'bar',              emphasis: {                focus: 'series'              },              data: [320, 332, 301, 334, 390, 330, 320]            },            {              name: 'Email',              type: 'bar',              stack: 'Ad',              emphasis: {                focus: 'series'              },              data: [120, 132, 101, 134, 90, 230, 210]            },            {              name: 'Union Ads',              type: 'bar',              stack: 'Ad',              emphasis: {                focus: 'series'              },              data: [220, 182, 191, 234, 290, 330, 310]            },            {              name: 'Video Ads',              type: 'bar',              stack: 'Ad',              emphasis: {                focus: 'series'              },              data: [150, 232, 201, 154, 190, 330, 410]            },            {              name: 'Search Engine',              type: 'bar',              data: [862, 1018, 964, 1026, 1679, 1600, 1570],              emphasis: {                focus: 'series'              },              markLine: {                lineStyle: {                  type: 'dashed'                },                data: [[{ type: 'min' }, { type: 'max' }]]              }            },            {              name: 'Baidu',              type: 'bar',              barWidth: 5,              stack: 'Search Engine',              emphasis: {                focus: 'series'              },              data: [620, 732, 701, 734, 1090, 1130, 1120]            },            {              name: 'Google',              type: 'bar',              stack: 'Search Engine',              emphasis: {                focus: 'series'              },              data: [120, 132, 101, 134, 290, 230, 220]            },            {              name: 'Bing',              type: 'bar',              stack: 'Search Engine',              emphasis: {                focus: 'series'              },              data: [60, 72, 71, 74, 190, 130, 110]            },            {              name: 'Others',              type: 'bar',              stack: 'Search Engine',              emphasis: {                focus: 'series'              },              data: [62, 82, 91, 84, 109, 110, 120]            }          ]        };        // 使用刚指定的配置项和数据显示图表        myEchart2.setOption(option2);      },    },  };  </script> |

**CoronavirusView.vue**

|  |
| --- |
| <template>    <div class="map">      <div id="mychart" style="width: 600px; height: 600px"></div>    </div>  </template>  <script>  import echarts from "echarts";  import "echarts/map/js/china";  import jsonp from "jsonp";  //ES6中let、const和var的区别,var定义的变量，作用域是整个封闭函数，是全域的；  //let定义的变量，作用域是在块级  //const用来专门声明一个常量，它跟let一样作用于块级作用域  let option = {    title: {      //标题      text: "中国实时疫情地图",      x: "center", //居中      textStyle: {        color: "#33539E",      },    },    tooltip: {      //提示信息      trigger: "item", //类型      //地图 : {a}（系列名称），{b}（区域名称），{c}（合并数值）, {d}（无）      formatter: "地区：{b}<br/>确诊：{c}",    },    series: [      //数据      {        type: "map", //图表的类型        map: "china",        label: {          //图形上的文本标签,用于显示数据信息          show: true,          color: "white",          fontSize: 10,        },        zoom: 1.3, //当前视角的缩放比例。        itemStyle: {          //地图区域的多边形,图形样式。          borderColor: "white",        },        emphasis: {          //高亮状态          label: {            //图形上的文本标签,用于显示数据信息            color: "#055",            fontSize: 12,          },          itemStyle: {            //地图区域的多边形 图形样式。            areaColor: "#A5678E",          },        },      },    ],    visualMap: {      //视觉地图，视觉映射组件      type: "piecewise", //分段型      show: true,      pieces: [        { min: 10000 }, // 不指定 max，表示 max 为无限大（Infinity）。        { min: 1000, max: 9999 },        { min: 100, max: 999 },        { min: 10, max: 99 },        { min: 1, max: 9 },        { value: 0 }, // 不指定 min，表示 min 为无限大（-Infinity）。      ],      inRange: {        //范围        color: ["#E8B7D4", "#BFB8DA", "#7FACD6", "#33539E"],      },      itemWidth: 10,      itemHeight: 10,    },    toolbox: {      show: true,      orient: "horizontal",      left: "right",      top: "top",      feature: {        dataView: { readOnly: false },        restore: {},        saveAsImage: {},      },    },  };  //指定模块默认输出一个data()函数，返回myEchart  export default {    name: "CoronavirusView.vue",    data() {      return {        myEchart: "",      };    },    mounted() {      // //钩子函数，生命周期为挂载完成后，Dom操作一般是在mounted钩子函数中进行。      this.getData();      // 基于准备好的dom，初始化echarts实例      this.myEchart = echarts.init(document.getElementById("mychart"));      // 使用刚指定的配置项和数据显示图表。      //this.myEchart.setOption(option);    },    methods: {      getData() {        //获取网络接口数据        //jsonp('url',function(){})        jsonp(          "https://interface.sina.cn/news/wap/fymap2020\_data.d.json",          (err, data) => {            // console.log(data.data.list)            //data.data.list            //使用js里面map()方法，循环遍历后，获取数据里面一部分值。            var lists = data.data.list.map((item) => {              return { name: item.name, value: item.value };            });            console.log(lists);            option.series[0].data = lists;            // 使用刚指定的配置项和数据显示图表。            this.myEchart.setOption(option);          }        );      },    },  };  </script>  <style scoped>  #mychart {    margin: 0 auto;  }  </style> |