### ECharts 数据可视化设计

#### 15.1 ECharts 图表展示

数据可视化就是把数据以更加直观的方式进行呈现,图表数据可视化的好处之一就是能 够清晰有效的传达信息和沟通信息,将数据以图表的方式呈现出来还可以帮助我们感受到那 些隐藏在数据之间的信息。

ECharts 是百度公司开源的一个使用 JavaScript 实现的开源可视化库,兼容性强,底层 依赖矢量图形库 ZRender,提供直观,交互丰富,可高度个性化定制的数据可视化图表。 ECharts 兼容当前绝大部分浏览器(IE8/9/10/11, Chrome, Firefox, Safari 等)及兼容多种 设备。ECharts 能够做出各种各样漂亮的图表,它能满足绝大多数可视化图表的实现,使用 方便、功能强大、是实现数据可视化的最佳选择之一,更多特点和介绍可以查阅官网地址: https://echarts.apache.org/zh/index.html。

1. ECharts 的安装

ECharts 的安装可以选择两种方式,一种是在页面中引入 echarts.js 文件,代码如下: <script src="js/echarts.min.js"></script>

若使用 Vue-cli (Vue 脚手架) 搭建项目,则可以使用语句 cnpm install echarts 进行安装, 若安装成功,在 node\_modules 路径下可以查看已安装的 ECharts 库,在项目终端窗口将显示 以下提示信息:

 $\sqrt{\text{Installed 1 packages}}$ 

 $\sqrt{\text{Linked 2 latest versions}}$ 

 $\sqrt{\text{Run 0 scripts}}$ 

 $\sqrt{\text{All packages installed (3 packages installed from npm registry, used 17s(network 17s),}$ speed 531.91kB/s, json 3(14.17kB), tarball 8.71MB)

2. ECharts 的使用和配置

安装完成后 ECharts 和 zrender 位于 node\_modules 目录下,我们可以直接在项目代码中 使用语句 require('echarts')来使用 ECharts。

(1) 在 main.js 文件添加以下代码,全局引入 ECharts 之后,即可在项目中使用 ECharts。 Import \* as echarts from 'echarts'

Vue.prototype.\$echarts = echarts

(2) 准备一个呈现图表的 div, 这个 div 的位置决定了图表的位置,实例如下:

<div id="container" style="width: 800px;height:800px;"></div>

(3) 初始化 ECharts 实例对象,指明图表显示哪一个 DOM 元素

var myChart = echarts.init(document.getElementById('container'))

(4)准备配置项: ECharts 图表有很多种类型,例如折线图、柱状图、饼图等,最终确 定显示的图表类型由配置项决定。对于不同的图表,基本是对配置项做设置,其他部分设置 不变。

15.1.1 折线图

通常一个 Echarts 图表通常由 title(标题)、legend(图例)、grid(网格)、xAxis(x轴)、yAxis (y轴)、dataZoom(区域缩放) tooltip(提示框组件)、toolbox(工具栏)、series(系列列 表,每个系列通过 type 决定图表类型)组成。

xAxis 代表直角坐标系中的 x 轴, 如果 type 属性的值为 category, 那么需要配置 data 数据, 代表在 x 轴的呈现; yAxis 代表直角坐标系中的 y 轴, 如果 type 属性配置为 value, 那 么无需配置 data, 此时 y 轴会自动去 series 下找数据进行图表的绘制; series 代表系列列表。每个系列通过 type 决定图表类型, data 用于设置每个系列的数据, 配置项都是以键值对的 形式存在。

各种类型的图表设计基础代码参见 ECharts 官网,本项目折线图表在浏览器中展示效果 如图 15-1 所示,代码如下。

<template>

<div>

<el-breadcrumb separator="/"

style="padding-left:10px;padding-bottom:10px;font-size:12px;">

<el-breadcrumb-item :to="{ path: '/main' }">首页</el-breadcrumb-item>

<el-breadcrumb-item>科研成果</el-breadcrumb-item>

<el-breadcrumb-item>知识产权</el-breadcrumb-item>

</el-breadcrumb>

<div ref="patentLine" id="patentLine" style="width:800px; height:600px">

</div>

</div>

</template>

<script>

```
type: 'value'
          },
          series: [{
             data: [15, 23, 38, 22, 40, 47, 73],
             type: 'line'
          }]
        }
     3
  },
  methods: {
     initChart () {
       const {
          patentLine,
          option
        \} = this
       patentLine.setOption(option)
     }
  },
  mounted () {
     this.patentLine = this.$echarts.init(this.$refs.patentLine)
     this.initChart()
  }
}
```

```
</script>
```



图 15-1 折线图表展示效果

### 15.1.2 柱状图

```
本项目折线图表在浏览器中展示效果如图 15-2 所示,代码如下。
<script>
export default {
  name: 'competition',
  data () {
    return {
    }
  },
  mounted(){
    this.drawLine();
  },
  methods: {
    drawLine(){
      // 基于准备好的 dom, 初始化 echarts 实例
      let competitionBar = this.$echarts.init(document.getElementById('competitionBar'))
      // 绘制图表
      competitionBar.setOption({
        title: { text: '学生参与各类竞赛获奖统计图' },
        tooltip: {},
        xAxis: {
          data: ["互联网+","计算机设计","挑战杯","领航杯","蓝桥杯","人工智能"]
        },
        yAxis: {},
        series: [{
          name: '获奖数量',
          type: 'bar',
          data: [16, 20, 36, 25, 40, 25]
        }]
      });
    }
  }
}
</script>
```



图 15-2 柱形图表展示效果

## 15.1.3 饼图

饼图的数据是由 name 和 value 组成的字典所形成的数组,饼图无须配置 xAxis 和 yAxis,例如以下代码可以设计一个饼图,展示效果如图 15-3 所示。

```
<script>
  export default {
  name: 'Paper',
  data () {
    return {
      paperPie: null,
      option: {
         title: { text: '部门各年度科研论文数量统计图' },
         series: [{
           data: [
              {name: 'SCI 检索',value: 21},
              {name: 'EI 检索',value: 47},
              {name: '中文核心', value: 13},
              {name: '普通期刊',value: 19},
              {name: '会议论文',value: 7}
           ],
           type: 'pie'
         }]
       }
    }
```

```
methods: {
    initChart () {
      const {
         paperPie,
         option
      = this
      paperPie.setOption(option)
    }
  },
  mounted () {
    // 两张引入方式都可以, 这里我用 ref,注意一定要节点 初始化完成
    // this.patentLine = this.$echarts.init(document.getElementById('patentLine'))
    this.paperPie = this.$echarts.init(this.$refs.paperPie)
    this.initChart()
  }
}
```

</script>



图 15-2 饼状图表展示效果

### 15.2 ECharts 地图图表展示

ECharts 展示地图形式图表主要有两种方式:一种是百度地图 API,需要在线申请使用 百度地图 API,实现在线联网展示地图;另外一种是矢量地图方式,通过矢量地图数据离线 展示地图。

以矢量地图的实现为例,实现方法如下:

(1) ECharts 基本的代码结构:

<!DOCTYPE html>

```
<html lang="en">
```

<head>

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

</head>

<body>

```
<div style="width: 800px;height:600px"></div>
```

<script>

```
var mCharts = echarts.init(document.querySelector("div"))
```

var option =  $\{\}$ 

```
mCharts.setOption(option)
```

</script>

```
</body>
```

</html>

```
(2)将地图的矢量 json 文件放到项目的 json/map/路径之下。
```

```
(3) 使用 Ajax 获取地图的 json 文件 "china.json", 代码如下:
```

```
$.get('json/map/china.json', function (chinaJson) {
```

```
})
```

```
(4) 在 Ajax 的回调函数中,向 ECharts 全局对象注册地图的 json 数据,代码如下:
```

```
$.get('json/map/china.json', function (chinaJson) {
```

```
echarts.registerMap('chinaMap', chinaJson)
```

```
})
```

```
(5) 获取到数据后,需要配置 geo 节点,再次设置 setOption
```

```
var mCharts = echarts.init(document.querySelector("div"))
```

```
$.get('json/map/china.json', function (chinaJson) {
```

```
echarts.registerMap('chinaMap', chinaJson)
```

```
var option = {
```

```
geo: {
```

```
type: 'map',// map 是一个固定的值
map: 'chinaMap',
```

```
//chinaMap 需要和 registerMap 中的第一个参数保持一致
```

```
};
```

}

```
mCharts.setOption(option)
```

```
})
```

# 15.3 ECharts 整合 Springboot 动态展示图表

本节介绍应用 Springboot 整合 ECharts 动态展示 MySQL 中的数据,如图 15-3 所示。





1. 配置项目后端数据访问接口文件 application.yml,代码如下:

server:

port: 1234

spring:

datasource:

driver-class-name: com.mysql.cj.jdbc.Driver

username: root

password: 123456

url:

jdbc:mysql://localhost:3306/echarts?useUnicode=true&characterEncoding=utf-8&allowMultiQue ries=true&useSSL=false&serverTimezone=GMT%2b8

type: com.alibaba.druid.pool.DruidDataSource

2. 在项目后端项目 entity 路径中添加 Echarts 实体类,代码如下:

package com.example.entity;

import com.baomidou.mybatisplus.annotation.IdType;

import com.baomidou.mybatisplus.annotation.TableId;

import com.baomidou.mybatisplus.annotation.TableName;

import com.fasterxml.jackson.annotation.JsonFormat;

import lombok.Data;

import java.util.Date;

@Data

@TableName("echarts")
public class Echarts {
 @TableId(value = "id", type = IdType.AUTO)
 private Long id;
 private String name;
 private Integer count;
 @JsonFormat(pattern = "yyyy-MM-dd", timezone = "GMT+8")
 private Date date;
}

}

3. 设计表示层 Controller,在 EchartsController.java 文件中新增 echarts 方法,即可完成 后端动态访问 ECharts 的设计,Vue 前端对饼图的设计参见第 7.1 节,代码如下:

package com.example.controller;

 $import\ com. baomidou. my batisplus. core. toolk it. Wrappers;$ 

import com.example.mapper.EchartsMapper;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import javax.annotation.Resource;

@RestController

@RequestMapping("/echarts")

public class EchartsController {

@Resource

private EchartsMapper echartsMapper;

@GetMapping

public Object echarts() {

// 查询数据库 echarts 表的所有数据

return echartsMapper.selectList(Wrappers.lambdaQuery());

}