链滴

使用 Vue+axios+Echarts 绘制关系图遇到的坑

作者：aopstudio
原文链接：https://ld246.com/article/1586781403456
来源网站：链滴

最近做毕设，后端采用的是Spring boot，通过Neo4jRepository从neo4j中获取数据，前端框架是Vu，通过axios请求后端数据，并处理成符合Echarts规范的数据格式，之后Echarts进行渲染。在前端渲数据的过程中遇到好多坑。

首先是如何对后端返回的数据进行处理。后端返回的数据格式是这样的：


```
[
  {
    "id": 1,
    "name": "CSS"
  },
  {
    "id": 2,
    "name": "Javascript",
    "derives": [
      {
        "id": 3,
        "name": "Vue.js"
      }
    ]
  },
  {
    "id": 3,
    "name": "Vue.js"
  },
  {
    "id": 171,
    "name": "前端技术",
    "includes": [
      {
        "id": 1,
        "name": "CSS"
      },
      {
        "id": 2,
        "name": "Javascript",
        "derives": [
          {
            "id": 3,
            "name": "Vue.js"
          }
        ]
      },
      {
        "id": 172,
        "name": "HTML",
      }
    ]
  },
  {
    "id": 172,
    "name": "HTML",
  }
]
```

对应的Echarts规范数据应当是：

```
data:[
  {
    name: "CSS"
  },
  {
    name: "Javascript"
  },
  {
    name: "Vue.js"
  },
  {
    name: "前端技术"
  },
  {
    name: "HTML"
  }
],
links: [
  {
    source: "Javascript",
    target: "Vue.js",
    name: "衍生出"
  },
  {
    source: "前端技术",
    target: "HTML",
    name: "包含"
  },
  {
    source: "前端技术",
    target: "CSS",
    name: "包含"
  },
  {
    source: "前端技术",
    target: "Javascript",
    name: "包含"
  }
]
```

遇到的坑主要是两个。第一个是处理返回的数据，返回数据存在对象resData中，我建立了两个空白组graphData和graphLinkes用于存储包装好的数据，然后使用for循环处理resData，正确的代码如：

```
for(var i=0,len=response.data.length;i<len;i++){
    me.graphData.push({
        name: me.resData[i].name,
        des: 'nodedes05',
        symbolSize: 50,
        category: 1,
    });
    if("includes" in me.resData[i]){
```

```
        let dataIncludes=me.resData[i].includes;
        for(var j=0,lenj=dataIncludes.length;j<lenj;j++){
            me.graphLinks.push({
                source: me.resData[i].name,
                target: dataIncludes[j].name,
                name: '包含',
                des: j
            })
        }
    }
    if("derives" in me.resData[i]){
        let dataDerives=me.resData[i].derives;
        for(var j=0,lenj=dataDerives.length;j<lenj;j++){
            me.graphLinks.push({
                source: me.resData[i].name,
                target: dataDerives[j].name,
                name: '衍生出',
                des: j
            })
        }
    }
}
```

但是一开始我想用for in循环却一直失败，不知道怎么回事。最后只能用最传统的i++。

第二个坑在Echarts渲染上。从后台获取数据的和渲染图形分别写在两个方法中，方法名为loadData drawline。一开始我是在mounted中先执行loadData，再执行drawline。但这样不能用。后来去网搜了一下说是在获取到数据后立马执行drawline才行，不过网上也没有说为什么。我想了一下这应该异步执行的问题。如果因为在loadData中使用的axios是异步方法，也就是不需要完全执行完里面的句就可以继续执行后面的方法。因此在我们调用drawline的时候数据根本还没有从后台接口获取到。如果在获取数据后立马执行，更确切的说是在axios then方法体内立马执行，而在axios方法内部的语还是同步执行的，这就保证了我们已经从后台接口获取到了数据，之后渲染就没问题了。

完整代码如下：

```
<template>

<div id="myChart"></div>
</template>

<style>
    #myChart{
        width: 100%;
        height: 1000px;
    }
</style>
<script>
export default {
    name:"Graph",
    data () {
    return {
      msg: 'Welcome to Your Vue.js App',
      categories: [],
      resData: [],
```

```
      graphData: [],
      graphLinks: []
    }
  },
  mounted(){
    for (var i = 0; i < 2; i++) {
        this.categories[i] = {
            name: '类目' + i
        };
    }
    this.loadData();
    //this.drawLine();
  },
  methods: {
    drawLine(){
        // 基于准备好的dom，初始化echarts实例
        let myChart = this.$echarts.init(document.getElementById('myChart'));
        // 绘制图表
        myChart.setOption({
            title: {
              text: 'ECharts 关系图'
            },
            tooltip: {},

            series: [{
              type: 'graph', // 类型:关系图
                layout: 'force', //图的布局，类型为力导图
                symbolSize: 40, // 调整节点的大小
                roam: true, // 是否开启鼠标缩放和平移漫游。默认不开启。如果只想要开启缩放或者平移
可以设置成 'scale' 或者 'move'。设置成 true 为都开启
                edgeSymbol: ['circle', 'arrow'],
                edgeSymbolSize: [2, 10],
                edgeLabel: {
                    normal: {
                        textStyle: {
                            fontSize: 20
                        }
                    }
                },
                force: {
                    repulsion: 2500,
                    edgeLength: [10, 50]
                },
                draggable: true,
                lineStyle: {
                    normal: {
                        width: 2,
                        color: '#4b565b',
                    }
                },
                edgeLabel: {
                    normal: {
                        show: true,
                        formatter: function (x) {
```

```
                    return x.data.name;
                }
            }
        },
        label: {
            normal: {
                show: true,
                textStyle: {}
            }
        },
        data: this.graphData,
        links: this.graphLinks,
        categories: this.categories,
    }]
});
},
loadData(){
    let me=this;
    axios.get('http://localhost:8080/graph/all')
    .then(function (response){
        me.resData=response.data;
        for(var i=0,len=response.data.length;i<len;i++){
            me.graphData.push({
                name: me.resData[i].name,
                des: 'nodedes05',
                symbolSize: 50,
                category: 1,
            });
            if("includes" in me.resData[i]){
                let dataIncludes=me.resData[i].includes;
                for(var j=0,lenj=dataIncludes.length;j<lenj;j++){
                    me.graphLinks.push({
                        source: me.resData[i].name,
                        target: dataIncludes[j].name,
                        name: '包含',
                        des: j
                    })
                }
            }
            if("derives" in me.resData[i]){
                let dataDerives=me.resData[i].derives;
                for(var j=0,lenj=dataDerives.length;j<lenj;j++){
                    me.graphLinks.push({
                        source: me.resData[i].name,
                        target: dataDerives[j].name,
                        name: '衍生出',
                        des: j
                    })
                }
            }
        }
        me.drawLine();
    })
    .catch(function (error) {
```

```
            console.log(error);
        });
    }
  }
}
</script>
```

原文链接：使用 Vue+axios+Echarts 绘制关系图遇到的坑