



链滴

修改 Markdown 编译器 vditor 支持 JS 形式 JSON，并支持非 JSON 形式 Option

作者: [zxniuniu](#)

原文链接: <https://ld246.com/article/1676024499285>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

Vditor 中的 echarts 图表，只能输入严格的 JSON 格式数据，参见 [Echarts页面渲染完整示例](#)，源码下 [Vditor 示例](#)，否则会报错，比如输入：

```
{
  backgroundColor: '#0e2147',
  grid: {
    left: '11%',
    top: '12%',
    right: '0%',
    bottom: '8%',
    containLabel: true
  },
  xAxis: [{
    show: false,
  }],
  yAxis: [{
    axisTick: 'none',
    axisLine: 'none',
    offset: '27',
    axisLabel: {
      textStyle: {
        color: '#ffffff',
        fontSize: '16',
      }
    },
  }],
  data: ['南昌', '广州', '杭州', '宁夏', '兰州', '南宁', '长沙', '武汉', '合肥', '贵州']
}, {
  axisTick: 'none',
  axisLine: 'none',
  axisLabel: {
    textStyle: {
      color: '#ffffff',
      fontSize: '16',
    }
  },
}, {
  data: ['10', '9', '8', '7', '6', '5', '4', '3', '2', '1']
}, {
  name: '分拨延误TOP 10',
  nameGap: '50',
  nameTextStyle: {
    color: '#ffffff',
    fontSize: '16',
  },
}, {
  axisLine: {
    lineStyle: {
      color: 'rgba(0,0,0,0)'
    }
  }
}, {
  data: [],
}, {
  series: [{
    name: '条',
    type: 'bar',
    yAxisIndex: 0,
```

```

data: [4, 13, 25, 29, 38, 44, 50, 52, 60, 72],
label: {
  normal: {
    show: true,
    position: 'right',
    textStyle: {
      color: '#ffffff',
      fontSize: '16',
    }
  }
},
barWidth: 12,
itemStyle: {
  normal: {
    color: function(params) {
      var myColor = ['#eb2100', '#eb3600', '#d0570e', '#d0a00e', '#34da62', '#00e9d',
', '#00c0e9', '#0096f3', '#33CCFF', '#33FFCC'];
      var num = myColor.length;
      return myColor[params.dataIndex % num]
    },
  },
},
z: 2
}, {
  name: '白框',
  type: 'bar',
  yAxisIndex: 1,
  barGap: '-100%',
  data: [99, 99.5, 99.5, 99.5, 99.5, 99.5, 99.5, 99.5, 99.5, 99.5],
  barWidth: 20,
  itemStyle: {
    normal: {
      color: '#0e2147',
      borderRadius: 5,
    }
  },
  z: 1
}, {
  name: '外框',
  type: 'bar',
  yAxisIndex: 2,
  barGap: '-100%',
  data: [100, 100, 100, 100, 100, 100, 100, 100, 100, 100],
  barWidth: 24,
  itemStyle: {
    normal: {
      color: function(params) {
        var myColor = ['#eb2100', '#eb3600', '#d0570e', '#d0a00e', '#34da62', '#00e9d',
', '#00c0e9', '#0096f3', '#33CCFF', '#33FFCC'];
        var num = myColor.length;
        return myColor[params.dataIndex % num]
      },
      borderRadius: 5,
    }
  }
}

```

```

    },
    z: 0
  },
  {
    name: '外圆',
    type: 'scatter',
    hoverAnimation: false,
    data: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    yAxisIndex: 2,
    symbolSize: 35,
    itemStyle: {
      normal: {
        color: function(params) {
          var myColor = ['#eb2100', '#eb3600', '#d0570e', '#d0a00e', '#34da62', '#00e9d',
            '#00c0e9', '#0096f3', '#33CCFF', '#33FFCC'];
          var num = myColor.length;
          return myColor[params.dataIndex % num]
        },
        opacity: 1,
      }
    },
    z: 2
  }
]
}

```

会报错：

echarts render error:

SyntaxError: Unexpected token s **in** JSON at position 2

只能输入以下严格的 JSON 格式才行，即所有字段以及字符串值都需要使用双引号括起来。

```

{
  "backgroundColor": "#0e2147",
  "grid": {
    "left": "11%",
    "top": "12%",
    "right": "0%",
    "bottom": "8%",
    "containLabel": true
  },
  "xAxis": [{
    "show": false
  }],
  "yAxis": [{
    "axisTick": "none",
    "axisLine": "none",
    "offset": "27",
    "axisLabel": {
      "textStyle": {
        "color": "#ffffff",
        "fontSize": "16"
      }
    }
  }
]
}

```

```

    },
    "data": ["南昌", "广州", "杭州", "宁夏", "兰州", "南宁", "长沙", "武汉", "合肥", "贵州"]
  }, {
    "axisTick": "none",
    "axisLine": "none",
    "axisLabel": {
      "textStyle": {
        "color": "#ffffff",
        "fontSize": "16"
      }
    },
    "data": ["10", "9", "8", "7", "6", "5", "4", "3", "2", "1"]
  }, {
    "name": "分拨延误TOP 10",
    "nameGap": "50",
    "nameTextStyle": {
      "color": "#ffffff",
      "fontSize": "16"
    },
    "axisLine": {
      "lineStyle": {
        "color": "rgba(0,0,0,0)"
      }
    },
    "data": []
  }],
  "series": [{
    "name": "条",
    "type": "bar",
    "yAxisIndex": 0,
    "data": [4, 13, 25, 29, 38, 44, 50, 52, 60, 72],
    "label": {
      "normal": {
        "show": true,
        "position": "right",
        "textStyle": {
          "color": "#ffffff",
          "fontSize": "16"
        }
      }
    }
  }],
  "barWidth": 12,
  "itemStyle": {
    "normal": {}
  },
  "z": 2
}, {
  "name": "白框",
  "type": "bar",
  "yAxisIndex": 1,
  "barGap": "-100%",
  "data": [99, 99.5, 99.5, 99.5, 99.5, 99.5, 99.5, 99.5, 99.5, 99.5],
  "barWidth": 20,
  "itemStyle": {

```

```

        "normal": {
            "color": "#0e2147",
            "barBorderRadius": 5
        }
    },
    "z": 1
}, {
    "name": "外框",
    "type": "bar",
    "yAxisIndex": 2,
    "barGap": "-100%",
    "data": [100, 100, 100, 100, 100, 100, 100, 100, 100, 100],
    "barWidth": 24,
    "itemStyle": {
        "normal": {
            "barBorderRadius": 5
        }
    },
    "z": 0
}, {
    "name": "外圆",
    "type": "scatter",
    "hoverAnimation": false,
    "data": [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    "yAxisIndex": 2,
    "symbolSize": 35,
    "itemStyle": {
        "normal": {
            "opacity": 1
        }
    },
    "z": 2
}]
}

```

最终生成图表如下：

```

var myColor = ['#eb2100', '#eb3600', '#d0570e', '#d0a00e', '#34da62', '#00e9db', '#00c0e9', '0096f3', '#33CCFF', '#33FFCC'];
option = {
    backgroundColor: '#0e2147',
    grid: {
        left: '11%',
        top: '12%',
        right: '0%',
        bottom: '8%',
        containLabel: true
    },
    xAxis: [{
        show: false,
    }],
    yAxis: [{
        axisTick: 'none',
        axisLine: 'none',
    }],
}

```

```

offset: '27',
axisLabel: {
  textStyle: {
    color: '#ffffff',
    fontSize: '16',
  }
},
data: ['南昌转运中心', '广州转运中心', '杭州转运中心', '宁夏转运中心', '兰州转运中心', '南宁转
中心', '长沙转运中心', '武汉转运中心', '合肥转运中心', '贵州转运中心']
}, {
  axisTick: 'none',
  axisLine: 'none',
  axisLabel: {
    textStyle: {
      color: '#ffffff',
      fontSize: '16',
    }
  },
  data: ['10', '9', '8', '7', '6', '5', '4', '3', '2', '1']
}, {
  name: '分拨延误TOP 10',
  nameGap: '50',
  nameTextStyle: {
    color: '#ffffff',
    fontSize: '16',
  },
  axisLine: {
    lineStyle: {
      color: 'rgba(0,0,0,0)'
    }
  },
  data: [],
}],
series: [{
  name: '条',
  type: 'bar',
  yAxisIndex: 0,
  data: [4, 13, 25, 29, 38, 44, 50, 52, 60, 72],
  label: {
    normal: {
      show: true,
      position: 'right',
      textStyle: {
        color: '#ffffff',
        fontSize: '16',
      }
    }
  },
  barWidth: 12,
  itemStyle: {
    normal: {
      color: function(params) {
        var num = myColor.length;
        return myColor[params.dataIndex % num]
      }
    }
  }
}]

```

```

    },
  },
  z: 2
}, {
  name: '白框',
  type: 'bar',
  yAxisIndex: 1,
  barGap: '-100%',
  data: [99, 99.5, 99.5, 99.5, 99.5, 99.5, 99.5, 99.5, 99.5, 99.5],
  barWidth: 20,
  itemStyle: {
    normal: {
      color: '#0e2147',
      barBorderRadius: 5,
    }
  },
  z: 1
}, {
  name: '外框',
  type: 'bar',
  yAxisIndex: 2,
  barGap: '-100%',
  data: [100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100],
  barWidth: 24,
  itemStyle: {
    normal: {
      color: function(params) {
        var num = myColor.length;
        return myColor[params.dataIndex % num]
      },
      barBorderRadius: 5,
    }
  },
  z: 0
},
{
  name: '外圆',
  type: 'scatter',
  hoverAnimation: false,
  data: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  yAxisIndex: 2,
  symbolSize: 35,
  itemStyle: {
    normal: {
      color: function(params) {
        var num = myColor.length;
        return myColor[params.dataIndex % num]
      },
      opacity: 1,
    }
  },
  z: 2
}
}

```



```
  ]  
};
```

但是目前官方还没修改, 参见 [希望echarts支持js格式的json](#) 其中提到, 通过 修改vditor的源码可实现, 把src\ts\markdown\chartRender.ts中的

```
const option = JSON.parse(text)
```

改成

```
var js = eval("(" + text + ")");  
var strJS = JSON.stringify(js);  
const option = JSON.parse(strJS)
```

即可实现, 但是有一点问题就是 **输入数据中不支持注释**。然后使用以下命令编译:

```
npm run build
```

编译成功后会生成dist目录, 使用下面两个文件

```
index.css  
index.min.js
```

替换vditor.css和vditor.js即可。

关于支持 JS 形式的配置项, 通过研究 Echarts 官网的 example-bundle.js 中涉及 JS 形式配置项的代码:

```
new Function("myChart", "app", "setTimeout", "setInterval", "ROOT_PATH", "var option; \n" +  
ext + "\nreturn option;")
```

发现, 可以采用 JS Function 来解决包含 JS 代码 的配置项结果获取, 即修改src\ts\markdown\charRender.ts中的

```
const option = JSON.parse(text)  
echarts.init(e, theme === "dark" ? "dark" : undefined).setOption(option);
```

为以下代码, 如果还有其他变量, 可以在最后一个参数前继续加入。

```
// 输入文本的函数定义, https://fuyiyi.imdo.co 如果有其他参数请加入  
var optionFun = new Function('myChart', 'setTimeout', 'setInterval', 'var option; \n' + text + '  
return option;');
```

```
// 原chartRender中的实例化后直接设置Option, 现在拆开  
var myChart = echarts.init(e, theme === "dark" ? "dark" : undefined);
```

```
// 通过原 JS 形式配置项 获取最终的option, https://fuyiyi.imdo.co  
const option = optionFun(myChart, setTimeout, setInterval);  
myChart.setOption(option);
```

然后编译后替换文件即可, 大功告成!

参考:

[修改vditor以支持javascript形式的json](#)

[vditor/chartRender.ts at master](#)

[charts/update-example.js snwowolf20170103/charts](#)

JavaScript 函数