目录

[1. 什么是Echarts 1](#_Toc426030119)

[2. 如何将数据交给Echarts 1](#_Toc426030120)

[2.1 模块化单文件引入 2](#_Toc426030121)

[2.2 标签式单文件引入 3](#_Toc426030122)

[3. 展现实例 6](#_Toc426030123)

[3.1数据准备 6](#_Toc426030124)

[3.2构建Option 6](#_Toc426030125)

[4. 如何自定义提示框 8](#_Toc426030126)

[5. 如何定制主题 9](#_Toc426030127)

**如何使用Echarts进行可视化展现**

**作者：王方舟**

# 什么是Echarts

Echarts缩写于Enterprise Charts，是一款开源、功能强大的商业级数据可视化产品，能够将数据更形象、更生动的通过可视化图表来展现。

该产品的原型是由百度前端研发人员根据百度内部业务需求开发的名为ZRender的轻量级Canvas库。之后百度成立了可视化团队，致力于轻量级可视化展现的研究，经过一系列的重构和丰富成为现在的Echarts产品。

现在Echarts具有较丰富的图标类型，支持基本图形：折线图、柱状图、饼图、散点图、环形图、雷达图等，以及复杂图形：地图、热力图、树图、和弦图等。

# 如何将数据交给Echarts

Echarts可视化产品的优势不仅仅体现在其丰富的图表类型支持，更体现在其简单的图表构建过程。Echarts提供了良好的调用接口，只需要构建一个包含一个<div>容器的html文本，引用指定的js文件，并书写简单的js语句将数据导入，即可进行可视化展现。

Echarts一共提供两种方式引入Echarts的js库，将分别于小节2.1及2.2介绍。

## 2.1 模块化单文件引入

此方式是Echarts官网推荐的一种方式，使用模块化单文件引入的好处是不需要加载全部的js代码，只需要加载指定图标类型的js文件即可。

以柱状图为例，模块化引入的代码如下图所示，简单的19行代码即实现了创建容器、引入echarts.js入口文件、配置加载路径等过程，但这只是模块化加载的第一步，也是不管使用哪种类型图表展现的必要过程。



图表 1 模块化单文件引入第一步

接着需引用柱状图的js文件，并将数据传给option参数，如图2所示。

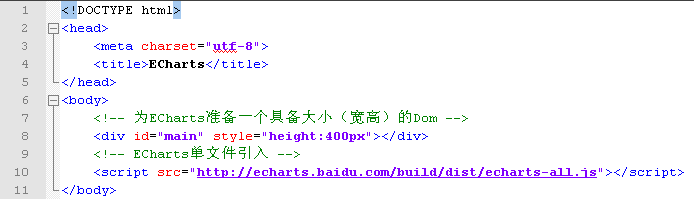


图表 2 模块化单文件引入第二步

第22、23行分别指定了引入echarts.js和bar.js（注此处不需要写.js），之后定义了function(ec)来实现数据的传递，首先定义myChart变量，进行初始化，再定义option变量，按照指定格式将图表的参数以json数组的形式传入，通过调用myChart.setOption(option)方式将json传递给图表对象。

## 2.2 标签式单文件引入

这种方式较模块化方式较简单，单文件顾名思义就是引入一个文件的意思，Echarts将所有的js实现文件全都放入了echart-all.js当中，直接引入该文件将可以展现所有类型的图表，如图3所示。



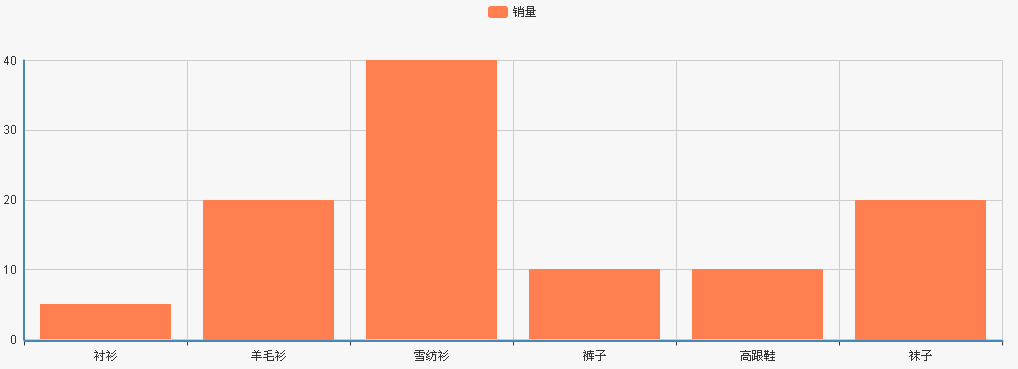
图表 3 单文件引入第一步

之后的操作类似于第一种方式，初始化图表以及传入option参数，如图4所示。



图表 4 单文件引入第二步

无论以两种哪种方式打开html文件，都会显示出如图5的柱状图。



图表 5 柱状图

# 展现实例

## 3.1数据准备

将数据可视化为图表的第一步是将数据读取到内存的数据结构当中。以折线图为例，需要准备的数据是X轴上的类别数组，以及Y轴上的数据二维数组（支持多个系列）。

其中将X轴上数据封装到xAxis对象中data[]数组中，而将Y轴上的数据封装到多个Series对象的data[]数组中。

xAxis : [

{

type : 'category',

data : ["衬衫","羊毛衫","雪纺衫","裤子","高跟鞋","袜子"]

}

],

series : [

{

"name":"销量",

"type":"bar",

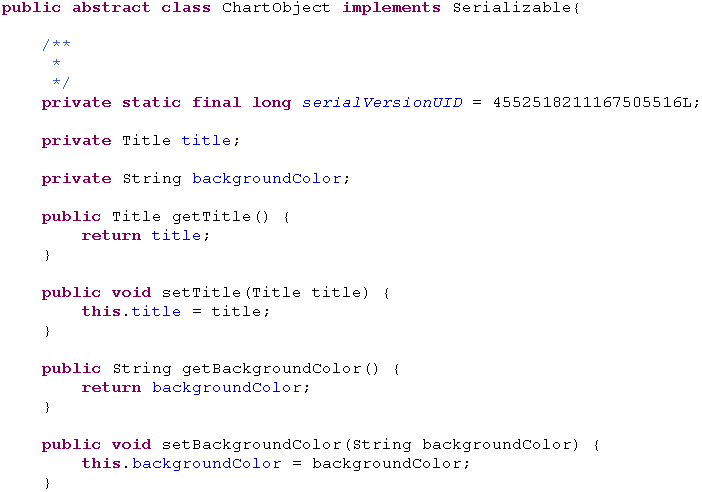
"data":[5, 20, 40, 10, 10, 20]

}

]

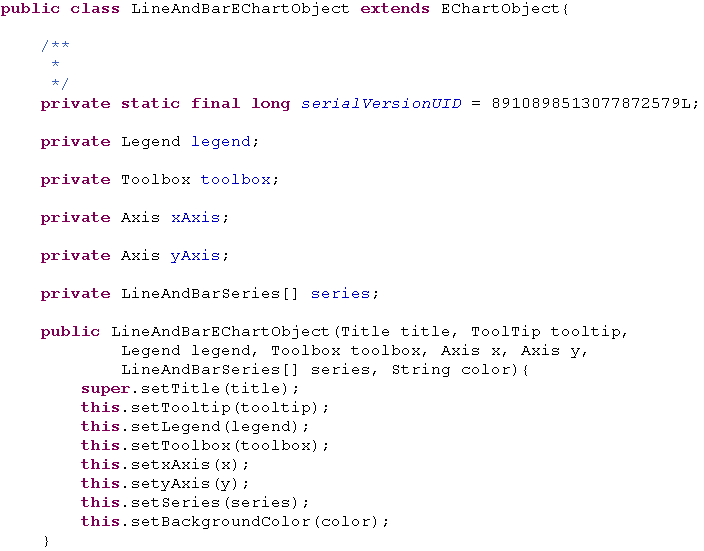
## 3.2构建Option

将数据准备好后只需要构建一个ChartObject的POJO对象，将图表标题、图例、XY轴、背景色等设置项设置完成后，通过jackson的ObjectMapper将对象转化为json数组。ChartObject的结构如图6所示。



图表 6 ChartObject

当然这只是一个抽象基类，由于每类图表所需的参数都各不相同，因此将所有图表的共同属性抽象为一层，即有具有标题和背景色。而折线图除了这两个属性外还具有图例、工具箱、X轴、Y轴及系列属性，如图7所示。



图表 7 折线图ChartObject

将每个属性的数据构造完成之后通过ObjectMapper将LineAndBarEcahrtObject对象转化为json，通过规定格式传给Option变量，拼接成如图2的HTML，既可以通过浏览器查看生成的图表。

# 如何自定义提示框

Echart自带的Tooltips提示是X/rY:value的形式，而如果想要在Tooltips中添加自定义的文字则需要以js写function来实现，tooltip属性具有参数formatter，用来控制tooltip的格式，例如书写如下所示的一段formatter：

"formatter":

function(params)

{

var res='聚合方式';

res+='<br/>'+params[0].seriesName+'='+params[0].data;

return res;

}

Formatter代表着内容格式器，支持异步回调，在折线图中其变量为：

{a}|{a0}：系列名称

{b}|{b0}：类目值

{c}|{c0}：数值

{d}|{d0}（部分图表类型无此项）：无

上述内容当中，定义了tooltip显示内容的格式为聚合方式\r系列名称=系列值，而非默认的X\rY:value，其中params对应着Series当中的内容，params[0].seriesName是系列名称，params[0].data是每行的系列值。

# 如何定制主题

Echarts官网上的实例具有调试的功能，左侧为图中可编辑的Option，右侧为图表展现，通过右侧下方的刷新按钮将文本编辑器中的Option放到Echarts中展现出相应的图表，而在刷新旁边同样提供了主题的切换，虽然在这里提供了14中主题，但在Echarts官网上下载的代码中就只有2中主题，分别为macarons和infographic，要实现这两种主题的加载只需要在init过程中传入该主题名称作为参数即可。如下所示：

var myChart = echarts.init(document.getElementById('main'), 'macarons');

若想在Echarts中加载其他主题，需要先使用审查元素功能，得到该主题的js文件，然后将该主题的定义放到echarts-all.js中。

例如blue主题定义如下：

n("echarts/theme/blue",[],function(){var e=

{

color:["#1790cf","#1bb2d8","#99d2dd","#88b0bb","#1c7099","#038cc4","#75abd0","#afd6dd"],

title:{textStyle:{fontWeight:"normal",color:"#1790cf"}},

dataRange:{color:["#1178ad","#72bbd0"]},

toolbox:{color:["#1790cf","#1790cf","#1790cf","#1790cf"]},

tooltip:{backgroundColor:"rgba(0,0,0,0.5)",axisPointer:{type:"line",lineStyle:{color:"#1790cf",type:'dashed'},crossStyle:{color:"#1790cf"},shadowStyle:{color:"rgba(200,200,200,0.3)"}}},

dataZoom:{dataBackgroundColor:"#eee",fillerColor:"rgba(144,197,237,0.2)",handleColor:"#1790cf"},grid:{borderWidth:0},

categoryAxis:{axisLine:{lineStyle:{color:"#1790cf"}},splitLine:{lineStyle:{color:["#eee"]}}},

valueAxis:{axisLine:{lineStyle:{color:"#1790cf"}},splitArea:{show:!0,areaStyle:{color:["rgba(250,250,250,0.1)","rgba(200,200,200,0.1)"]}},splitLine:{lineStyle:{color:["#eee"]}}},

timeline:{lineStyle:{color:"#1790cf"},controlStyle:{normal:{color:"#1790cf"},emphasis:{color:"#1790cf"}}},

gauge:{axisLine:{show:!0,lineStyle:{color:[[.2,"#1bb2d8"],[.8,"#1790cf"],[1,"#1c7099"]],width:8}},axisTick:{splitNumber:10,length:12,lineStyle:{color:"auto"}},splitLine:{length:18,lineStyle:{color:"auto"}},pointer:{length:90%,color:"auto"},

title:{textStyle:{color:"#333"}},detail:{textStyle:{color:"auto"}}},textStyle:{fontFamily:"微软雅黑, Arial, Verdana, sans-serif"}};return e})

上面的代码定义了蓝色的主题，其中定义的内容分别为：颜色版、标题、数据范围、提示框、数据缩进、类别轴、值轴、时间轴、通用部分、全局字体等，将该主题定义到echarts.js后，需要将init方法改为：

var myChart = echarts.init(document.getElementById('main'), 'blue');

生成的图表将按照新的主题进行配色。