**Echarts**

## 一、基础知识

**1 ECharts是什么**

ECharts是由百度团队开发的，可高度个性化定制的数据可视化图表库。它的底层依赖轻量级的Canvas类库ZRender，是一个纯JavaScript的图标库，兼容（IE8/9/10/11，Chrome，Firefox，Safari等）主流浏览器，可以运行在PC和移动设备上。

**2 ECharts是纯Javascript图表库**

基于Canvas，底层依赖ZRender，商业产品常用图表库，提供直观，生动，可交互，可个性化定制的数据可视化图表。创新的拖拽重计算、数据视图、值域漫游等特性大大增强了用户体验，赋予了用户对数据进行挖掘、整合的能力。图表类型支持**折线图（区域图）、柱状图（条状图）、散点图（气泡图）、K线图、饼图（环形图）、雷达图（填充雷达图）、和弦图、力导向布局图、地图**，同时支持任意维度的堆积和多图表混合展现。本次主要和大家交流一下我用echarts制作世界地图的一些经验和心得，以及在制作过程遇到的问题和解决方法。

**3 丰富的可视化类型**

ECharts 提供了常规的折线图、柱状图、散点图、饼图、K线图，用于统计的盒形图，用于地理数据可视化的地图、热力图、线图，用于关系数据可视化的关系图、treemap、旭日图，多维数据可视化的平行坐标，还有用于 BI 的漏斗图，仪表盘，并且支持图与图之间的混搭。

除了已经内置的包含了丰富功能的图表，ECharts 还提供了自定义系列，只需要传入一个renderItem函数，就可以从数据映射到任何你想要的图形，更棒的是这些都还能和已有的交互组件结合使用而不需要操心其它事情。你可以在下载界面下载包含所有图表的构建文件，如果只是需要其中一两个图表，又嫌包含所有图表的构建文件太大，也可以在在线构建中选择需要的图表类型后自定义构建。

**4 多种数据格式无需转换直接使用**

ECharts 内置的 dataset 属性（4.0+）支持直接传入包括二维表，key-value 等多种格式的数据源，通过简单的设置 encode 属性就可以完成从数据到图形的映射，这种方式更符合可视化的直觉，省去了大部分场景下数据转换的步骤，而且多个组件能够共享一份数据而不用克隆。为了配合大数据量的展现，ECharts 还支持输入 TypedArray 格式的数据，TypedArray 在大数据量的存储中可以占用更少的内存，对 GC 友好等特性也可以大幅度提升可视化应用的性能。

**5 千万数据的前端展现**

通过增量渲染技术（4.0+），配合各种细致的优化，ECharts 能够展现千万级的数据量，并且在这个数据量级依然能够进行流畅的缩放平移等交互。几千万的地理坐标数据就算使用二进制存储也要占上百 MB 的空间。因此 ECharts 同时提供了对流加载（4.0+）的支持，你可以使用 WebSocket 或者对数据分块后加载，加载多少渲染多少！不需要漫长地等待所有数据加载完再进行绘制。

**6 移动端优化**

ECharts 针对移动端交互做了细致的优化，例如移动端小屏上适于用手指在坐标系中进行缩放、平移。 PC 端也可以用鼠标在图中进行缩放（用鼠标滚轮）、平移等。细粒度的模块化和打包机制可以让 ECharts 在移动端也拥有很小的体积，可选的 SVG 渲染模块让移动端的内存占用不再捉襟见肘。

**7 多渲染方案，跨平台使用！**

ECharts 支持以 Canvas、SVG（4.0+）、VML 的形式渲染图表。VML 可以兼容低版本 IE，SVG 使得移动端不再为内存担忧，Canvas 可以轻松应对大数据量和特效的展现。不同的渲染方式提供了更多选择，使得 ECharts 在各种场景下都有更好的表现。除了 PC 和移动端的浏览器，ECharts 还能在 node 上配合 node-canvas 进行高效的服务端渲染（SSR）。从 4.0 开始我们还和微信小程序的团队合作，提供了 ECharts 对小程序的适配！社区热心的贡献者也为我们提供了丰富的其它语言扩展，比如 Python 的pyecharts，R 语言的 recharts, Julia 的 ECharts.jl 等等。

**8 深度的交互式数据探索**

交互是从数据中发掘信息的重要手段。“总览为先，缩放过滤按需查看细节”是数据可视化交互的基本需求。ECharts 一直在交互的路上前进，我们提供了 图例、视觉映射、数据区域缩放、tooltip、数据刷选等开箱即用的交互组件，可以对数据进行多维度数据筛取、视图缩放、展示细节等交互操作。

**9 官方实例**

**<http://echarts.baidu.com/examples/#chart-type-bar>**

## 二、简单实例

**1.引入ECharts**

ECharts3开始不再强制使用AMD的方式按需引入，那么简单粗暴，直接在官网下载所需ECharts文件，像引入普通JavaScript库一样用script标签引入即可。简单吧，看代码：

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>ECharts入门</title>

</head>

<body>

<!-- 引入ECharts文件 -->

<script src='echarts.min.js'></script>

</body>

</html>

注：建议把js文件放在body下方

**2.准备一个具备宽高的DOM容器**

<body>

<!-- 为ECharts准备一个具备大小（宽高）的DOM容器-->

<div id='main' style='width:600px;height:400px;'></div>

</body>

**3.初始化一个echarts实例**

<body>

<!-- 为ECharts准备一个具备大小（宽高）的DOM容器-->

<div id='main' style='width:600px;height:400px;'></div>

<!--引入echarts.js-->

<script src='./echarts.min.js'></script>

<script>

//基于准备好的DOM，初始化echarts实例

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

</script>

</body>

**4.指定图表的配置项和数据**

var option = {

title:{

text:'EChars入门'

},

//提示框组件

tooltip:{

//坐标轴触发，主要用于柱状图，折线图等

trigger:'axis'

},

//图例

legend:{

data:['销量']

},

//横轴

xAxis:{

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

},

//纵轴

yAxis:{},

//系列列表。每个系列通过type决定自己的图表类型

series:[{

name:'销量',

//折线图

type:'line',

data:[5, 20, 36, 10, 10, 20]

}]

};

**5.显示图标**

myChart.setOption(option);

**6.完整代码**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>ECharts入门</title>

</head>

<body>

<!-- 为ECharts准备一个具备大小（宽高）的DOM容器-->

<div id='main' style='width:600px;height:400px;'></div>

<!-- 引入ECharts文件 -->

<script src='./echarts.min.js'></script>

<script>

//基于准备好的DOM，初始化echarts实例

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

//指定图表的配置项和数据

var option = {

title:{

text:'EChars入门'

},

//提示框组件

tooltip:{

//坐标轴触发，主要用于柱状图，折线图等

trigger:'axis'

},

//图例

legend:{

data:['销量']

},

//横轴

xAxis:{

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

},

//纵轴

yAxis:{},

//系列列表。每个系列通过type决定自己的图表类型

series:[{

name:'销量',

//折线图

type:'line',

data:[5, 20, 36, 10, 10, 20]

}]

};

//使用刚指定的配置项和数据显示图表

myChart.setOption(option);

</script>

</body>

</html>

**7.生成图表**

