

Table of Contents

Introduction	1.1
ECharts介绍	1.2
引入ECharts	1.3
静态加载数据显示K线图	1.4
动态加载数据显示K线图	1.5



主要内容

这里主要是介绍基于开源项目ECharts来开发的K线图，介绍了ECharts的基本的特性和基本使用方法，以及我们演示如何去开发k线图和动态数据去显示K线图，我们围绕着ECharts来构建K线图涉及到的问题来介绍。

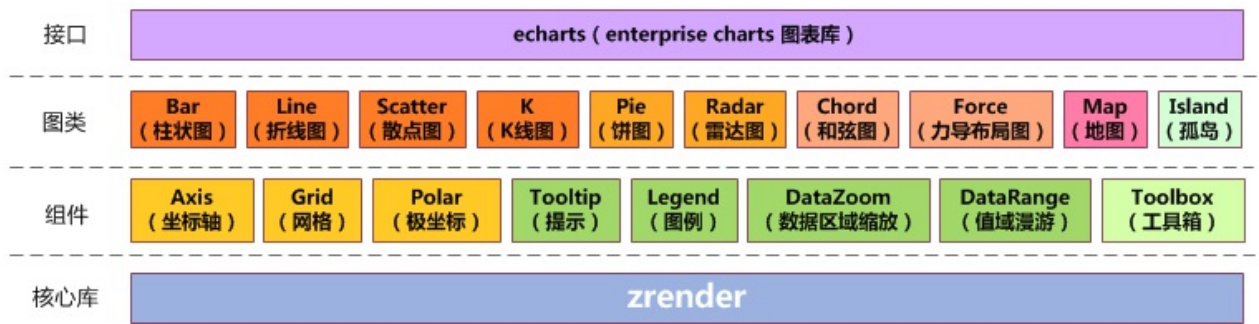
ECharts简介

ECharts，缩写来自**Enterprise Charts**，来自百度商业前端数据可视化团队，基于html5 Canvas，是一个纯JavaScript图表库，可以流畅运行在PC和移动设备上，兼容当前绝大部分浏览器（IE6/7/8/9/10/11，chrome，firefox，Safari等），底层依赖轻量级的Canvas类库**ZRender**，提供直观，生动，可交互，可高度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。

ECharts 3 中更是加入了更多丰富的交互功能以及更多的可视化效果，并且对移动端做了深度的优化，本文介绍的是最新的3.0版本重点介绍的是如何来绘制K线图。

支持的图表类型

ECharts具有坐标系，图例，提示，工具箱等基础组件，并在此上构建出折线图（区域图）、柱状图（条状图）、散点图（气泡图）、饼图（环形图）、K线图、地图、和弦图以及力导向布局图，同时支持任意维度的堆积和多图表混合展现。



常用组件

名称	描述
title	标题，每个图表最多仅有一个标题控件，每个标题控件可设主副标题。
xAxis	直角坐标系 grid 中的 x 轴
yAxis	直角坐标系 grid 中的 y 轴
Grid	直角坐标系中除坐标轴外的绘图网格
Legend	图例组件展现了不同系列的标记(symbol)，颜色和名字。可以通过点击图例控制哪些系列不显示。
Tooltip	提示框组件
Toolbox	工具栏。内置有导出图片，数据视图，动态类型切换，数据区域缩放，重置五个工具。

涉及的图表

名称	描述
Line	折线图，堆积折线图，区域图，堆积区域图
Bar	柱形图（纵向），堆积柱形图，条形图（横向），堆积条形图
candlestick	K线图，蜡烛图

获取 ECharts

你可以通过以下几种方式获取 ECharts。

1. 从[官网下载界面](#)选择你需要的版本下载，根据开发者功能和体积上的需求，我们提供了不同打包的下载，如果你在体积上没有要求，可以直接下载[完整版本](#)。开发环境建议下载[源代码版本](#)，包含了常见的错误提示和警告。
2. 在 ECharts 的[GitHub](#)上下载最新的 `release` 版本，解压出来的文件夹里的 `dist` 目录里可以找到最新版本的 echarts 库。
3. 通过 npm 获取 echarts，`npm install echarts --save`，详见“[在 webpack 中使用 echarts](#)”
4. cdn 引入，你可以在[cdnjs](#)，[npmcdn](#)或者国内的[bootcdn](#)上找到 ECharts 的最新版本。

模块化单文件引入（推荐）

```
<!--从当前页面，引入模块加载器esl.js-->
<script src="esl.js"></script>
```

```
<!-- 为ECharts准备一个具备大小（宽高）的Dom -->
<div id="main" style="height:400px"></div>
```

```

// 为模块加载器配置echarts的路径，从当前页面链接到echarts.js，定义所需的图表路径
require.config({
  paths:{
    'echarts' : './echarts',
    'echarts/chart/bar' : './echarts'
  }
});

// 使用
require(
[
  'echarts',
  'echarts/chart/bar' // 使用柱状图就加载bar模块，按需加载
],
function (ec) {
  // 基于准备好的dom，初始化echarts图表
  var myChart = ec.init(document.getElementById('main'));

  var option = {
    tooltip: {
      show: true
    },
    legend: {
      data:['销量']
    },
    xAxis : [
      {
        type : 'category',
        data : ["衬衫","羊毛衫","雪纺衫","裤子","高跟鞋","袜子"]
      }
    ],
    yAxis : [
      {
        type : 'value'
      }
    ],
    series : [
      {
        "name":"销量",
        "type":"bar",
        "data":[5, 20, 40, 10, 10, 20]
      }
    ]
  };

  // 为echarts对象加载数据
  myChart.setOption(option);
}
);

```

标签式单文件引入

```
<!-- ECharts单文件引入 -->
<script src="echarts.js"></script>
```

```
<!-- 为ECharts准备一个具备大小（宽高）的Dom -->
<div id="main" style="height:400px"></div>
```

```
// 基于准备好的dom，初始化echarts图表
var myChart = echarts.init(document.getElementById('main'));

var option = {
  tooltip: {
    show: true
  },
  legend: {
    data:['销量']
  },
  xAxis : [
    {
      type : 'category',
      data : ["衬衫","羊毛衫","雪纺衫","裤子","高跟鞋","袜子"]
    }
  ],
  yAxis : [
    {}
  ],
  series : [
    {
      "name": "销量",
      "type": "bar",
      "data": [5, 20, 40, 10, 10, 20]
    }
  ]
};

// 为echarts对象加载数据
myChart.setOption(option);
```

自定义构建echarts单文件

ECharts支持自定义构建echarts文件，我们可以在官网的[Builder](#)页面，勾选上我们需要的组件和图表类型，以及是否支持IE8和压缩文件，生成一个js文件，引用即可使用。

静态数据示例

在这里我们拿一个demo来演示如何开发K线图，在demo中一次性加载数据，并在demo中显示k线并同时显示折线和柱状线，k线主要是针对开、关、高、低数据进行绘制的；而折现是针对MA5和DIF和DEA指标参数进行绘制的，MACD指标参数是利用柱状图进行绘制的。

效果



主要过程

```
<!--从当前页面，引入模块加载器es1.js-->
```

```
<script src="es1.js"></script>
```

```
<!--为ECharts准备一个具备大小（宽高）的Dom -->
```

```
<div id="main" style="height:600px; width:1000px;"></div>
```



```
// 为模块加载器配置echarts的路径，从当前页面链接到echarts.js，定义所需的图表路径
require.config({
  paths:{
    'echarts' : './echarts',
    'echarts/chart/bar' : './echarts',//引入柱状图
    'echarts/chart/line' : './echarts',//引入折线图
    'echarts/chart/candlestick' : './echarts',//引入K线图
  }
});
```

```
// 使用
require([
  'echarts',
  'echarts/chart/bar', // 使用柱状图就加载bar模块，按需加载
  'echarts/chart/line',
  'echarts/chart/candlestick'
],
function (ec) {

  //数据模型 time0 open1 close2 min3 max4 vol5 tag6 macd7 dif8 dea9
  //['2015-10-19',18.56,18.25,18.19,18.56,55.00,0,-0.00,0.08,0.09]
  var data = splitData([
    ['2015-10-16',18.4,18.58,18.33,18.79,67.00,1,0.04,0.11,0.09],
    ['2015-10-19',18.56,18.25,18.19,18.56,55.00,0,-0.00,0.08,0.09],
    ['2015-10-20',18.3,18.22,18.05,18.41,37.00,0,0.01,0.09,0.09],
    ['2015-10-21',18.18,18.69,18.02,18.98,89.00,0,0.03,0.10,0.08],
    ['2015-10-22',18.42,18.29,18.22,18.48,43.00,0,-0.06,0.05,0.08],
    ['2015-10-23',18.26,18.19,18.08,18.36,46.00,0,-0.10,0.03,0.09],
    ['2015-10-26',18.33,18.07,17.98,18.35,65.00,0,-0.15,0.03,0.10],
    ['2015-10-27',18.08,18.04,17.88,18.13,37.00,0,-0.19,0.03,0.12],
    ['2015-10-28',17.96,17.86,17.82,17.99,35.00,0,-0.24,0.03,0.15],
    ['2015-10-29',17.85,17.81,17.8,17.93,27.00,0,-0.24,0.06,0.18],
    ['2015-10-30',17.79,17.93,17.78,18.08,43.00,0,-0.22,0.11,0.22],
    ['2015-11-02',17.78,17.83,17.78,18.04,27.00,0,-0.20,0.15,0.25],
    ['2015-11-03',17.84,17.9,17.84,18.06,34.00,0,-0.12,0.22,0.28],
    ['2015-11-04',17.97,18.36,17.85,18.39,62.00,0,-0.00,0.30,0.30],
    ['2015-11-05',18.3,18.57,18.18,19.08,177.00,0,0.07,0.33,0.30],
    ['2015-11-06',18.53,18.68,18.3,18.71,95.00,0,0.12,0.35,0.29],
    ['2015-11-09',18.75,19.08,18.75,19.98,202.00,1,0.16,0.35,0.27],
    ['2015-11-10',18.85,18.64,18.56,18.99,85.00,0,0.09,0.29,0.25],
    ['2015-11-11',18.64,18.44,18.31,18.64,50.00,0,0.06,0.27,0.23],
    ['2015-11-12',18.55,18.27,18.17,18.57,43.00,0,0.05,0.25,0.23],
    ['2015-11-13',18.13,18.14,18.09,18.34,35.00,0,0.05,0.24,0.22],
    ['2015-11-16',18.01,18.1,17.93,18.17,34.00,0,0.07,0.25,0.21],
    ['2015-11-17',18.2,18.14,18.08,18.45,58.00,0,0.11,0.25,0.20],
    ['2015-11-18',18.23,18.16,18.0,18.45,47.00,0,0.13,0.25,0.19],
    ['2015-11-19',18.08,18.2,18.05,18.25,32.00,0,0.15,0.24,0.17],
    ['2015-11-20',18.15,18.15,18.11,18.29,36.00,0,0.13,0.21,0.15],
```

```

        ['2015-11-23', 18.16, 18.19, 18.12, 18.34, 47.00, 0, 0.11, 0.18, 0.13],
        ['2015-11-24', 18.23, 17.88, 17.7, 18.23, 62.00, 0, 0.03, 0.13, 0.11],
        ['2015-11-25', 17.85, 17.73, 17.56, 17.85, 66.00, 0, -0.03, 0.09, 0.11],
        ['2015-11-26', 17.79, 17.53, 17.5, 17.92, 63.00, 0, -0.10, 0.06, 0.11],
        ['2015-11-27', 17.51, 17.04, 16.9, 17.51, 67.00, 0, -0.16, 0.05, 0.13],
        ['2015-11-30', 17.07, 17.2, 16.98, 17.32, 55.00, 0, -0.12, 0.09, 0.15]
    ]);
    //数组处理
    function splitData(rawData) {
        var datas = [];
        var times = [];
        var vols = [];
        var macds = []; var difs = []; var deas = [];
        for (var i = 0; i < rawData.length; i++) {
            datas.push(rawData[i]);
            times.push(rawData[i].splice(0, 1)[0]);
            vols.push(rawData[i][4]);
            macds.push(rawData[i][6]);
            difs.push(rawData[i][7]);
            deas.push(rawData[i][8]);
        }
        return {
            datas: datas,
            times: times,
            vols: vols,
            macds: macds,
            difs: difs,
            deas: deas
        };
    }

    //分段计算
    function fenduans(){
        var markLineData = [];
        var idx = 0; var tag = 0; var vols = 0;
        for (var i = 0; i < data.times.length; i++) {
            //初始化数据
            if(data.datas[i][5] != 0 && tag == 0){
                idx = i; vols = data.datas[i][4]; tag = 1;
            }
            if(tag == 1){ vols += data.datas[i][4]; }
            if(data.datas[i][5] != 0 && tag == 1){
                markLineData.push([
                    xAxis: idx,
                    yAxis: data.datas[idx][1]>data.datas[idx][0]?(data.datas[idx][3]).toFixed(2):(data.datas[idx][2]).toFixed(2),
                    value: vols
                ], {
                    xAxis: i,
                    yAxis: data.datas[i][1]>data.datas[i][0]?(data.datas[i][3]).toFixed(2):(data.datas[i][2]).toFixed(2)
                });
            }
        }
    }

```

```

    ]]);
    idx = i; vols = data.datas[i][4]; tag = 2;
  }

  //更替数据
  if(tag == 2){ vols += data.datas[i][4]; }
  if(data.datas[i][5] != 0 && tag == 2){
    markLineData.push([
      xAxis: idx,
      yAxis: data.datas[idx][1]>data.datas[idx][0]?(data.datas[idx][3]).toFixed(2):(data.datas[idx][2]).toFixed(2),
      value: (vols/(i-idx+1)).toFixed(2)+' M'
    ], {
      xAxis: i,
      yAxis: data.datas[i][1]>data.datas[i][0]?(data.datas[i][3]).toFixed(2):(data.datas[i][2]).toFixed(2)
    });
    idx = i; vols = data.datas[i][4];
  }
}
return markLineData;
}

//MA计算公式
function calculateMA(dayCount) {
  var result = [];
  for (var i = 0, len = data.times.length; i < len; i++) {
    if (i < dayCount) {
      result.push('-');
      continue;
    }
    var sum = 0;
    for (var j = 0; j < dayCount; j++) {
      sum += data.datas[i - j][1];
    }
    result.push((sum / dayCount).toFixed(2));
  }
  return result;
}

var option = {
  title: {
    text: 'K线周期图表',
    left: 0
  },
  tooltip: {
    trigger: 'axis',
    axisPointer: {
      type: 'line'
    }
  },
},

```

```

legend: {
  data: ['KLine', 'MA5']
},
grid: [
  {
    left: '3%',
    right: '1%',
    height: '60%'
  }, {
    left: '3%',
    right: '1%',
    top: '71%',
    height: '10%'
  }, {
    left: '3%',
    right: '1%',
    top: '82%',
    height: '14%'
  }
],
xAxis: [{
  type: 'category',
  data: data.times,
  scale: true,
  boundaryGap: false,
  axisLine: { onZero: false },
  splitLine: { show: false },
  splitNumber: 20,
  min: 'dataMin',
  max: 'dataMax'
}, {
  type: 'category',
  gridIndex: 1,
  data: data.times,
  axisLabel: {show: false}
}, {
  type: 'category',
  gridIndex: 2,
  data: data.times,
  axisLabel: {show: false}
}],
yAxis: [{
  scale: true,
  splitArea: {
    show: false
  }
}, {
  gridIndex: 1,
  splitNumber: 3,
  axisLine: {onZero: false},
  axisTick: {show: false},
  splitLine: {show: false},
  axisLabel: {show: true}
}

```

```

}, {
  gridIndex: 2,
  splitNumber: 4,
  axisLine: {onZero: false},
  axisTick: {show: false},
  splitLine: {show: false},
  axisLabel: {show: true}
}],
dataZoom: [{
  type: 'inside',
  xAxisIndex: [0, 0],
  start: 20,
  end: 100
}, {
  show: true,
  xAxisIndex: [0, 1],
  type: 'slider',
  top: '97%',
  start: 20,
  end: 100
}, {
  show: false,
  xAxisIndex: [0, 2],
  type: 'slider',
  start: 20,
  end: 100
}],
series: [
  // 数据部分，驱动图表生成的数据内容数组，数组中每一项为一个系列的选项及数据
  {
    name: 'K线周期图表',
    type: 'candlestick',
    data: data.datas,
    itemStyle: {
      normal: {
        color: '#ef232a',
        color0: '#14b143',
        borderColor: '#ef232a',
        borderColor0: '#14b143'
      }
    },
    {
      name: 'MA5',
      type: 'line',
      data: calculateMA(5),
      smooth: true,
      lineStyle: {
        normal: {
          opacity: 0.5
        }
      }
    }
  }, {
    name: 'Volumn',
    type: 'bar',

```

```

        xAxisIndex: 1,
        yAxisIndex: 1,
        data: data.vols,
        itemStyle: {
            normal: {
                color: function(params) {
                    var colorList;
                    if (data.datas[params.dataIndex][1]>data.datas[params.dataI
ndex][0]) {
                        colorList = '#ef232a';
                    } else {
                        colorList = '#14b143';
                    }
                    return colorList;
                },
            }
        }
    },{
        name: 'MACD',
        type: 'bar',
        xAxisIndex: 2,
        yAxisIndex: 2,
        data: data.macds,
        itemStyle: {
            normal: {
                color: function(params) {
                    var colorList;
                    if (params.data >= 0) {
                        colorList = '#ef232a';
                    } else {
                        colorList = '#14b143';
                    }
                    return colorList;
                },
            }
        }
    },{
        name: 'DIF',
        type: 'line',
        xAxisIndex: 2,
        yAxisIndex: 2,
        data: data.difs
    },{
        name: 'DEA',
        type: 'line',
        xAxisIndex: 2,
        yAxisIndex: 2,
        data: data.deas
    }
]
};

```

```
// 基于准备好的dom，初始化echarts实例  
var myChart = ec.init(document.getElementById('main'));  
  
myChart.setOption(option);  
});
```

动态加载数据

在这里我们拿一个demo来演示如何动态加载数据来显示K线图，在demo中通过websocket来接收服务端发送过来的数据，并在demo中显示k线的时候同时显示折线，k线主要是针对开、关、高、低数据进行绘制的；而折线是针对MA5、MA10、MA20和MA30指标参数进行绘制。

效果



主要过程

```
<!--加载echarts库-->  
<script src="echarts.min.js"></script>
```

```
<!--为ECharts准备一个具备大小（宽高）的Dom -->  
<div id="main" style="height:600px; width:1000px;"></div>
```

```
<!-- websocket客户端连接-->  
var wsUri = "ws://127.0.0.1:7777";  
  
function testWebSocket()  
{  
    var websocket = new WebSocket(wsUri);  
    websocket.onopen = function(evt) { onOpen(evt) };  
    websocket.onclose = function(evt) { onClose(evt) };  
    websocket.onmessage = function(evt) { onMessage(evt) };  
    websocket.onerror = function(evt) { onError(evt) };  
}  
  
function onOpen(evt)
```



```

{
    console.log("CONNECTED");
}

function onClose(evt)
{
    console.log("DISCONNECTED");
}

function calculateMA(id,dayCount) {
    var data0= option.series[0].data;
    if(data0.length >= dayCount)
    {
        var sum=0;
        for (var i = 0; i < dayCount; i++) {
            sum += parseFloat(data0[data0.length -1 - i][1]);
        }
        option.series[id].data.push((sum / dayCount).toFixed(2));
    }
    else
    {
        option.series[id].data.push('-');
    }
}

/*
*接收到的客户端数据，解析并计算，重新绘制k线图
*数据模型 time open close min max
*[2010/1/4,3289.75,3243.76,3243.319,3295.279]
*/
function onMessage(evt)
{
    var rawData = evt.data.split(',');

    option.xAxis.data.push(rawData.splice(0, 1)[0]);
    option.series[0].data.push(rawData);
    calculateMA(1,5);
    calculateMA(2,10);
    calculateMA(3,20);
    calculateMA(4,30);
    myChart.setOption(option);//重绘K线图
}

function onError(evt)
{
    console.log('ERROR:' + evt.data);
}

window.addEventListener("load", testWebSocket, false);

```

```

<!-- echarts去绘制图-->
///WEBSOCKET RECEIVE
// 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
// 数据意义：开盘(open)，收盘(close)，最低(lowest)，最高(highest)
var option = {
  animation: false,
  title: {
    text: '上证指数',
    left: 0
  },
  tooltip: {
    trigger: 'axis',
    formatter: function(params) {
      var res = params[0].name
      res += '<br/> 开盘 : ' + params[0].data[0] + ' 最高 : ' + params[0].data[3]
      res += '<br/> 收盘 : ' + params[0].data[1] + ' 最低 : ' + params[0].data[2]
      return res;
    },
    axisPointer: {
      type: 'line',
      animation: false
    }
  },
  legend: {
    data: ['日K', 'MA5', 'MA10', 'MA20', 'MA30']
  },
  toolbox: {
    show : true,
    feature : {
      mark : {show: true},
      dataView : {
        show: true,
        readOnly: true
      },
      restore : {show: true},
      saveAsImage : {show: true}
    }
  },
  grid: {
    left: '10%',
    right: '10%',
    bottom: '15%'
  },

```

```

xAxis: {
  type: 'category',
  data: [],
  scale: true,
  boundaryGap : true,
  axisLine: { lineStyle: { color: '#8392A5' } ,onZero: false},
  splitLine: {show: false},
  splitNumber: 20,
  min: 'dataMin',
  max: 'dataMax'
},
yAxis: {
  scale: true,
  axisLine: { lineStyle: { color: '#8392A5' }},
  splitLine: { show: true }
},
dataZoom: [
  {
    type: 'inside',
  },
  {
    show: true,
    type: 'slider',
    y: '90%',
  }
],
series: [
  {
    name: '日K',
    type: 'candlestick',
    data: [],
    itemStyle: {
      normal: {
        color: '#FD1050',
        color0: '#0CF49B',
        borderColor: '#FD1050',
        borderColor0: '#0CF49B'
      }
    }
  },
  {
    name: 'MA5',
    type: 'line',
    data: [],
    smooth: true,
    lineStyle: {
      normal: {opacity: 0.5}
    }
  }
],

```

```

    {
      name: 'MA10',
      type: 'line',
      data: [],
      smooth: true,
      lineStyle: {
        normal: {opacity: 0.5}
      }
    },
    {
      name: 'MA20',
      type: 'line',
      data: [],
      smooth: true,
      lineStyle: {
        normal: {opacity: 0.5}
      }
    },
    {
      name: 'MA30',
      type: 'line',
      data: [],
      smooth: true,
      lineStyle: {
        normal: {opacity: 0.5}
      }
    }
  ]
};

```

```
myChart.showLoading();
```

```
// 使用刚指定的配置项和数据显示图表。
```

```
var start = new Date();
```

```
myChart.setOption(option);
```

```
console.log(new Date() -start + 'ms');
```

```
myChart.hideLoading();
```