# HightChart详细使用设计

# 配置

|  |
| --- |
| Highcharts.setOptions({  global:{  useUTC:**false**  },lang:{  resetZoom:'复 位',  shortMonths:['01','02','03','04','05','06','07','08','09','10','11','12'],  weekdays: ['星期日', '星期一', '星期二', '星期三', '星期四', '星期五', '星期六']  }  }); |

# 曲线spline

|  |
| --- |
| var alarmLineChart={  chart: {  type: 'spline'  ,backgroundColor:'#303031'  },  title: {  text: '告警指标曲线走势图',  style:{'color':'#6D869F','fontSize':'16px','fontWeight':'bold'}  },  subtitle: {  text: ''  },  legend:{  align: 'right',  layout: 'vertical',  verticalAlign:'middle',  padding:10,  backgroundColor:'#454E5A',  lineHeight:20,  itemStyle:{color: '#CCCCCC','fontSize':'10px'},  itemHiddenStyle:{color: '#3A3A3A'}  },  tooltip: {  backgroundColor:'#2E2E2F',  borderColor:'',  borderRadius:4,//默认3  hideDelay:1000,//默认500  valueDecimals:1,//默认保留所有小数,不使用formatter函数时有效  valuePrefix:'',//不使用formatter函数时有效  valueSuffix:'',//不使用formatter函数时有效  style:{color: 'white',fontSize: '14px',padding: '5px',fontFamily:'微软雅黑',height:300},  formatter: function() {  //Highcharts.dateFormat('%e. %b', this.x)  return '<b>'+ this.series.name +'</b><br/>'+  this.x +'-> '+ this.y ;  }  },  plotOptions: {  spline: {  lineWidth: 2,  states: {  hover: {  lineWidth: 3  }  },  // dataLabels:{  // enabled :true,//是否在点的旁边显示数据  // rotation:0  // },  marker: {  enabled: false,  radius:2//点的半径  }  }  },  exporting: {  enabled: false  },  // colors:['#2B908F','#434348','#7CB5EC'],  colors:['#BEC3F2','#90EE7E','#7CB5EC'],  xAxis: {  // type: 'datetime',  // alternateGridColor:'',  // gridLineWidth:0,  categories:[],  crosshair: true,  labels:{  distance: 15,//轴标签距绘图区边界，默认 15px  align: "center",  // enabled: false,//是否显示  enabled:true,  rotation:0,//轴坐标旋转角度，默认 0  // style:{color: '#6D869F',fontFamily:'微软雅黑',fontSize:'12px'}  style:{color: '#C0D0E0',fontFamily:'微软雅黑',fontSize:'12px'}  },  lineWidth:1,  lineColor:'#C0D0E0'  // minorGridLineColor:'red',  // minorGridLineWidth:5,  // minorTickLength:1,  // tickWidth:2  // type:string 默认linear采用图表数据点的名称做分类名称 linear", "logarithmic", "datetime" 或者 "category"      },  yAxis: [  {  title: {  text: '排队数',  style:{color: '#7CB5EC',fontFamily:'微软雅黑',fontSize:'12px'}  },  // alternateGridColor:'#656565',//相间的网格行颜色  // ceiling:2,//允许最高的自动计算的坐标轴刻度极端值；  min: 0,  gridLineColor:'#6D869F',//网格线的颜色  gridLineWidth:1,//设置为0时不显示  labels:{  style:{color: '#7CB5EC',fontFamily:'微软雅黑',fontSize:'12px'}  },  lineColor:'#6D869F',//刻度线的颜色  lineWidth:0,//轴线宽度，默认是0  tickWidth:1//刻度线的宽度，默认是0  },{  title: {  text: '平均响应时间(s)',  style:{color: '#90EE7E',fontFamily:'微软雅黑',fontSize:'12px'}  },  // alternateGridColor:'#656565',//相间的网格行颜色  // ceiling:2,//允许最高的自动计算的坐标轴刻度极端值；  min: 0,  gridLineColor:'#6D869F',//网格线的颜色  gridLineWidth:1,//设置为0时不显示  labels:{  style:{color: '#90EE7E',fontFamily:'微软雅黑',fontSize:'12px'}  },  lineColor:'#6D869F',//刻度线的颜色  lineWidth:0,//轴线宽度，默认是0  tickWidth:1//刻度线的宽度，默认是0  },    {  title: {  text: '请求数',  style:{color: '#BEC3F2',fontFamily:'微软雅黑',fontSize:'12px'}  },  // alternateGridColor:'#656565',//相间的网格行颜色  // ceiling:2,//允许最高的自动计算的坐标轴刻度极端值；  min: 0,  gridLineColor:'#6D869F',//网格线的颜色  gridLineWidth:1,//设置为0时不显示  labels:{  style:{color: '#BEC3F2',fontFamily:'微软雅黑',fontSize:'12px'}  },  lineColor:'#6D869F',//刻度线的颜色  lineWidth:0,//轴线宽度，默认是0  tickWidth:1//刻度线的宽度，默认是0  }  ],    series: [  {  name: '请求数',  yAxis: 0,  data: []    }, {  name: '平均响应时间',  yAxis: 1,  data: []  }, {  name: '排队数',  yAxis: 2,  data: []  }  ]  }; |

## 请求数据

|  |
| --- |
| $.post("/iptvSupervise/currentAlarm/getIndexData.do",{startTime:startTime,dimenIp:dimen\_ip},**function**(data){    alarmLineChart.xAxis.categories = data.Xaxis  alarmLineChart.series[0].data = data.reqcounts;  alarmLineChart.series[1].data = data.avgres;  alarmLineChart.series[2].data = data.queencounts;  $('#alarmLineDrawingChart\_div').highcharts(alarmLineChart);  }); |

# 屏幕分屏

|  |
| --- |
| **var** gridHeiht=$("body").height();  // var gridHeiht=window.screen.height;  **var** gridWidth=window.screen.width;  **var** win = Ext.create('Ext.panel.Panel', {  layout: 'column',  bodyStyle:{margin:'0px',padding:'0px'},  width:gridWidth,  items: [{  columnWidth:.3333333,  items:[  {html:'<div id="container" style="height:'+gridHeiht/3+'px;width:'+gridWidth/3+'px;"></div>'},  {html:'<div id="container2" style="height:'+gridHeiht/3+'px;width:'+gridWidth/3+'px;"></div>'},  {html:'<div id="container3" style="height:'+gridHeiht/3+'px;width:'+gridWidth/3+'px;"></div>'}  ]  }, {  columnWidth:.3333333,  items:[  {html:'<div id="container4" style="height:'+gridHeiht/3+'px;width:'+gridWidth/3+'px;"></div>'},  {html:'<div id="container5" style="height:'+gridHeiht/3+'px;width:'+gridWidth/3+'px;"></div>'},  {html:'<div id="container6" style="height:'+gridHeiht/3+'px;width:'+gridWidth/3+'px;"></div>'}  ]  }, {  columnWidth:.3333333,  items:[  {html:'<div id="container7" style="height:'+gridHeiht/3+'px;width:'+(gridWidth-(gridWidth/3\*2))+'px;"></div>'},  {html:'<div id="container8" style="height:'+gridHeiht/3+'px;width:'+(gridWidth-(gridWidth/3\*2))+'px;"></div>'},  {html:'<div id="container9" style="height:'+gridHeiht/3+'px;width:'+(gridWidth-(gridWidth/3\*2))+'px;"></div>'}  ]  }],  renderTo: Ext.getBody()  });  $('#container').highcharts(chart);  win.doLayout(); |

# 时间格式化

**Highcharts**.dateFormat(**'%Y/%m/%e %H:%M'**, **this**.x)

%a: 简短型星期,比如‘Mon’.

%A: 完整型星期, 比如‘Monday’

%d: 两位的日期, 从01到31.

%e: 数字型日期,从 1 到 31.

%b: 简短型月份, 比如 ‘Jan’.

%B: 完整型月份, 比如 ‘January’.

%m: 两位的月份, 从01 到 12.

%y: 两位数的年份, 比如 09 代表 2009.

%Y: 四位数的年份, 比如 2009.

%H: 两位数的24小时制的时间, 从00 到 23.

%I: 两位输得12小时制的时间, 从 00 到 11.

%l (小写字母l):12小时制的时间, 1 到 11.

%M: 两位的分钟, 从00 到 59.

%p: 小写形式的 AM 或 PM.

%P: 大写形式的 AM or PM.

%S: 两位的秒, 从00 到 59

# y轴设置固定刻度

yAxis: {

title: {

text: name + "(" + end + ")"

},

tickPositions: [10, 20, 30,40,50,60,70,80,90,100]

},

# tooltip

|  |
| --- |
| tooltip: {  shared:**true**,  crosshairs: **true**,  formatter: **function** () {  **var** tip = '<b>' + Highcharts.dateFormat('%Y/%m/%d', **this**.x) + '</b><br>';  $.each(**this**.points, **function**(index, el) {  **if**(el.series.name=='语音rab成功次数') {  tip += '<span style="color: ' + el.color + ';">' + el.series.name + ': ' + el.y.toFixed(0) + '</span><br>';  }**else** **if**(el.series.name=='语音rab用户数'){  tip+= '<span style="color: ' + el.color + ';">' + el.series.name + ': ' + el.y.toFixed(0) + '</span><br>';  }**else**{  tip+= '<span style="color: ' + el.color + ';">' + el.series.name + ': ' + el.y.toFixed(2) + ' %</span><br>';  }  });  **return** tip;  }  } |

# 柱状图

## 堆叠图

|  |
| --- |
| **var** chart = {  chart: {  type: 'column'  },  title: {  text: '区局L网'  },  colors:['#89CDEF','#FFC57E','#434348'],  xAxis: {  type: 'category',  labels:{  // style:{color:'#3B3B3B'}  }  },  yAxis: [{ lineWidth: 1,  tickWidth:2,  gridLineWidth: 1,  allowDecimals: **false**,  min: 0,  title: {  text: '',  style: {  color: Highcharts.getOptions().colors[0]  }  }  }, { // Tertiary yAxis  lineWidth: 1,  tickWidth:2,  gridLineWidth: 1,  title: {  text: '',  style: {  color: Highcharts.getOptions().colors[1]  }  },  labels: {  format: '{value} %',  style: {  color: Highcharts.getOptions().colors[1]  }  },  opposite: **false**  }],  tooltip: {  shared: **true**,  crosshairs: **true**,  formatter: **function**() {    **var** tip = '<b>区局L网</b><br>',total='',rate='';  $.each(**this**.points, **function**(index, el) {  **if**(el.series.name=='室分'||el.series.name=='宏站') {  tip += '<span style="color: ' + el.color + ';">' + el.series.name + ': ' + el.y + '</span><br>';  total=**this**.point.stackTotal;  }**else**{  rate= '<span style="color: ' + el.color + ';">' + el.series.name + ': ' + el.y + ' %</span><br>';  }  });  tip+='基站总数: ' + total+'<br>';  tip+=rate;  **return** tip;  }  },  plotOptions: {  column: {  stacking: 'normal'  },  spline:{  cursor:'pointer',  lineWidth:2,  marker:{  radius:0  }  }  },  series: []  };  $('#chart2').highcharts(chart);  **var** chartObj=$('#chart2').highcharts();  chartObj.showLoading();  $.getJSON('/mapv/drawChart/getLNetWorkData.do',{},**function**(data){  chart.series=data;  $('#chart2').highcharts(chart);  chartObj.hideLoading();  })  }  //数据形式  Map dataMap=**new** HashedMap();  dataMap.put("name", map.get("region"));  dataMap.put("y", Integer.*parseInt*(v[1]));  Map data\_a=**new** HashedMap();  data\_a.put("name", "宏站");  data\_a.put("data", data\_0);  Map data\_b=**new** HashedMap();  data\_b.put("name", "室分");  data\_b.put("data", data\_1);  Map data\_c=**new** HashedMap();  data\_c.put("name", "RPB下行利用率");  data\_c.put("type", "spline");  data\_c.put("yAxis", 1);  data\_c.put("data", data\_3); |