Highcharts之动态刷新——结合后台数据

时间 2015-05-27 14:05:45  [ITeye-博客](http://www.tuicool.com/sites/jaeeAn)

原文  [http://raising.iteye.com/blog/2214779](http://raising.iteye.com/blog/2214779?utm_source=tuicool&utm_medium=referral)

主题 [Highcharts](http://www.tuicool.com/topics/11060097)

最近要求做前台的一些东西，用到了Highcharts。。其实在Highcharts中文网上有相应的演示demo，一般你所需要的图的样式都可以找到。难点在于实际开发中，需要的数据往往是要结合后台请求并响应的数据的。

附Highcharts中文网：

http://www.hcharts.cn/demo/index.php

而我这次做的是趋势图，涉及到动态刷新，做的过程还是花了一番功夫的，也补充和巩固了一点js的知识，为了纪念，把过程记录一下：

首先，是引入HIghcharts绘图相关的js文件和jQuery.js。

接下来，把HIghcharts动态刷新的格式拷贝并作修改：

$(**function**(){

$(document).ready(**function**() {

Highcharts.setOptions({

global: {

useUTC: false//是否使用世界标准时间

}

});

**var** chart;

chart = **new** Highcharts.Chart({

chart: {

renderTo: 'container',

type: 'spline',

marginRight: 10,

events: {

load: **function**() {

**var** series = **this**.series;

setInterval(**function**() {

**var** result = reload();

**var** x = result.time;

**for**(**var** i=0; i<series.length; i++) {

**var** y = result.y[i];

series[i].addPoint([x, y], true, true);

}

}, 1000\*5);

}

}

},

title: {

text: 'ssssss'

},

xAxis: {

type: 'datetime',

tickPixelInterval: 150

},

yAxis: {

title: {

text: 'Value'

},

plotLines: [{

value: 0,

width: 1,

color: '#808080'

}]

},

tooltip: {

formatter: **function** () {

**return** '<b>' + **this**.series.name + '</b><br/>' +

Highcharts.dateFormat('%Y-%m-%d %H:%M:%S', **this**.x) + '<br/>' +

Highcharts.numberFormat(**this**.y, 2);

}

},

//图例属性

legend: {

layout: 'vertical',

align: 'right',

verticalAlign: 'top',

borderWidth: 0

},

exporting: {

enabled: false

},

series: create()

});

});

});

重点是chart里面的event属性，series属性。注意，此时：series属性是一个js函数的返回值，需用“js函数名()”来取得。

而demo里面，series是这样的：

series: [{

name: 'Random data',

data: (**function**() {

// generate an array of random data

**var** data = [],

time = (**new** Date()).getTime(),

i;

**for** (i = -19; i <= 0; i++) {

data.push({

x: time + i \* 1000,

y: Math.random()

});

}

**return** data;

})()

}]

这是死数据，并且只显示一条曲线（或折线）。我们需要得到后台数据，并且x,y的值都需要由后台决定。

create()方法的定义如下：

**function** **create**() {

**var** series = **new** Array();

$.ajax({

type: "POST",

url: "xxxx/yyyyy.json",

async: false, //表示同步，如果要得到ajax处理完后台数据后的返回值，最好这样设置

success: **function**(result){

**var** channels = result.key;

**var** size = channels.length;

**for**(**var** i=0; i<size; i++) {

**var** name = channels[i].yyyy;

**var** perTotalCnt = channels[i].xxxx;

**var** data = **function**() {

**var** data = [],

time = result.time,

i;

**for**(i=-6; i<=0; i++) {

data.push({

x: aaaa,

y: zzzz

});

}

**return** data;

}();

series.push({"name": name, "data": data});

}

}

}, false); //false表示“遮罩”，前台不显示“请稍后”进度提示

alert(series);

**return** series;

}

其中， series.push({"name": name, "data": data});这一行就是在往series数组[]中push数据，从而生成格式中需要的series数组；

而这一部分：

events: {

load: **function**() {

**var** series = **this**.series;

setInterval(**function**() {

**var** result = reload();

**var** x = result.time;

**for**(**var** i=0; i<series.length; i++) {

**var** y = result.y[i];

series[i].addPoint([x, y], true, true);

}

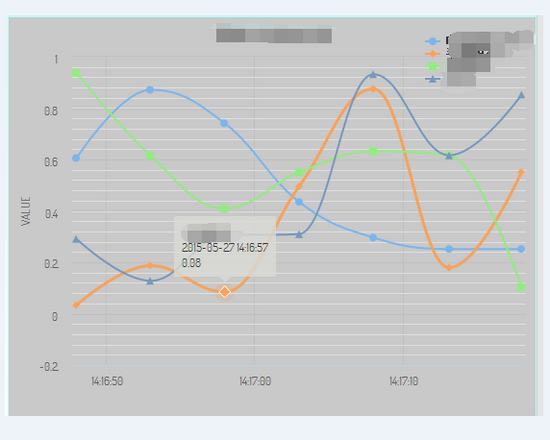
}, 1000\*5);

}

}

则是每隔一段时间(1000\*5=5s)刷新页面数据，并且通过循环生成多条"曲线"。其中reload()也是一个js函数的返回值：注：定义json对象的方法var json = {"key1": value1, "key2":value2} (reload方法就不贴了，也是通过ajax请求后台，得到相应数据并且处理相应数据后，放到一个数据结构或json对象中)。

效果：



画图完毕。