

## 图表插件 echarts 涟漪效果 effectScatter 添加配置项

Echart 的组件的默认效果是这样的：



那么，问题来了，我可能需要老版本那样的东西像泡一样的的图：



我查过相关的问题，别人给出的解答是 3 版本之后放弃了这一个效果，所以，下面列出我在最新的版本（3.36）中，如何用 effectScatter 效果来达到预测的效果：

首先我们找到生成联谊效果的位置：

```

47225
47226
47227     var zrUtil = __webpack_require__(4);
47228     var symbolUtil = __webpack_require__(107);
47229     var graphic = __webpack_require__(43);
47230     var numberUtil = __webpack_require__(7);
47231     var Symbol = __webpack_require__(106);
47232     var Group = graphic.Group;
47233     //涟漪效果的波纹数
47234     var EFFECT_RIPPLE_NUMBER = 3;
47235

```

上面的注释是我加的，正常情况应该在 47288-47234 行之间，如果嫌麻烦可以直接搜 EFFECT\_RIPPLE\_NUMBER，好了，但是听我说，echart 读取配置项的方法（setOption(option)），和 canvas 的生成方法不是分开的，这就意味着即使你找到这一行，也并没有多大的作用，你在这个方法里面不能通过加入配置项的方法来随时改变涟漪的波纹数，所以，需要继续往下找：

源码是这样的：

```

47280     effectSymbolProto.startEffectAnimation = function (effectCfg) {
47281         var symbolType = effectCfg.symbolType;
47282         var color = effectCfg.color;
47283         var rippleGroup = this.childAt(1);
47284
47285         for (var i = 0; i < EFFECT_RIPPLE_NUMBER; i++) {
47286             // var ripplePath = symbolUtil.createSymbol(
47287             //     symbolType, -0.5, -0.5, 1, 1, color
47288             // );
47289             // If width/height are set too small (e.g., set to 1) on ios10
47290             // and macOS Sierra, a circle stroke become a rect, no matter what
47291             // the scale is set. So we set width/height as 2. See #4136.
47292             var ripplePath = symbolUtil.createSymbol(
47293                 symbolType, -1, -1, 2, 2, color
47294             );
47295             ripplePath.attr({

```

这个方法是向 canvas 中画涟漪效果的方法，EFFECT\_RIPPLE\_NUMBER 就是刚刚找到的那个预先定义的变量，修改之后我希望他能够读取我的写的配置，而不仅仅只有 API 中的，就是扩展 echart 的 API：

首先看 effectScatter 的 API：

```

    {
      type: 'effectScatter',
      name: ...,
      legendHoverLink: true,
      effectType: 'ripple',
      showEffectOn: 'render',
      rippleEffect: {
        period: 4,
        scale: 2.5,
        brushType: 'fill',
      },
      coordinateSystem: 'cartesian2d',
      xAxisIndex: 0,
      yAxisIndex: 0,
      polarIndex: 0,
      geoIndex: 0,
      symbol: 'circle',
      symbolSize: 10
    }
  ]
}

```

RippleEffect 项中的参数只有三个，这里我希望能都传入第四个参数 num 作为涟漪的个数，就象这样：

```

rippleEffect: {
  num: 3,
  period: 2,
  scale: 10,
  brushType: 'fill'
},

```

实际情况是，即使你的配置（option）传入 echarts 的方法之后，方法在解析 option 的时候，依然会重新构造一次，因此，不是说我加了一项 num，他就能理所应当的读出这个配置，

```

47392     var effectCfg = {};
47393
47394     effectCfg.showEffectOn = seriesModel.get('showEffectOn');
47395     effectCfg.rippleScale = itemModel.get('rippleEffect.scale');
47396     effectCfg.brushType = itemModel.get('rippleEffect.brushType');
47397     effectCfg.period = itemModel.get('rippleEffect.period') * 1000;
47398     effectCfg.effectOffset = idx / data.count();
47399     effectCfg.z = itemModel.getShallow('z') || 0;
47400     effectCfg.zlevel = itemModel.getShallow('zlevel') || 0;
47401     effectCfg.symbolType = symbolType;
47402     effectCfg.color = color;
47403

```

这里就需要构造我们所加的配置：

```

47397
47398     var effectCfg = {};
47399
47400     effectCfg.showEffectOn = seriesModel.get('showEffectOn');
47401     effectCfg.rippleScale = itemModel.get('rippleEffect.scale');
47402     effectCfg.brushType = itemModel.get('rippleEffect.brushType');
47403     //构造 rippleEffect-num 配置项
47404     effectCfg.num = itemModel.get('rippleEffect.num');
47405     effectCfg.period = itemModel.get('rippleEffect.period') * 1000;
47406     effectCfg.effectOffset = idx / data.count();
47407     effectCfg.z = itemModel.getShallow('z') || 0;
47408     effectCfg.zlevel = itemModel.getShallow('zlevel') || 0;
47409     effectCfg.symbolType = symbolType;
47410     effectCfg.color = color;
47411
47412     this.off('mouseover').off('mouseout').off('emphasis').off('normal');
47413
47414     if (effectCfg.showEffectOn === 'render') {
47415         this._effectCfg
47416             ? this.updateEffectAnimation(effectCfg)
47417             : this.startEffectAnimation(effectCfg);
47418
47419         this._effectCfg = effectCfg;

```

然后在那个 for 循环那里加入传入的配置 num:

```

effectSymbolProto.startEffectAnimation = function (effectCfg) {
    console.log(effectCfg);
    var symbolType = effectCfg.symbolType;
    var color = effectCfg.color;
    var rippleGroup = this.childAt(1);
    var rippleNumber = effectCfg.num;

    //这里舍弃从EFFECT_RIPPLE_NUMBER 47234行传入的固定参数，改为从option取值，将渲染数量做成配置
    for (var i = 0; i < rippleNumber; i++) {
        // var ripplePath = symbolUtil.createSymbol(
        //     symbolType, -0.5, -0.5, 1, 1, color
        // );
        // If width/height are set too small (e.g., set to 1) on ios10
        // and macOS Sierra, a circle stroke become a rect, no matter what
        // the scale is set. So we set width/height as 2. See #4136.
        var ripplePath = symbolUtil.createSymbol(
            symbolType, -1, -1, 2, 2, color
        );
    }
};

```

别忘了还有这里:

```

        z2: 99,
        silent: true,
        scale: [0.5, 0.5]
    });

    var delay = -i / rippleNumber * effectCfg.period + effectCfg.effectOffset;
    // TODO Configurable effectCfg.period
    ripplePath.animate('', true)
        .when(effectCfg.period, {
            scale: [effectCfg.rippleScale / 2, effectCfg.rippleScale / 2]
        })
        .delay(delay)

```

到这里基本就完成添加配置项的任务，当然，在 min 版本中，哦我们需要改这几个地方：  
首先搜索 startEffectAnimation:

```
$.removeAll(),f.startEffectAnimation=function(t){for(var e=t.symbolType,i=t.color,n=this.childAt(1),r=2
$./*第一处，获取传进来的参数t effectCfg/q= t.num;r<q;r++){var a=s.createSymbol(e,-1,-1,2,2,i):a.attr2
$.({style:{strokeNoScale:!0},z2:99,silent:!0, scale:[.5,.5]});/*第二处 传入的参数修改*/var l=-r/q*t.period+t.effectOffset,a2
$.animate("").when(t.period,{scale:[t.rippleScale/2,t.rippleScale/2]}).delay(1).start(),a.animateStyle(!0).when(t2
$.period,{opacity:0}).delay(1).start(),n.add(a)}o(n,t)),f.updateEffectAnimation=function(t){for(var e=this._effectCfg,2
$.i=this.childAt(1),n=["symbolType","period","rippleScale"],r=0;r<n;r++){var a=n[r]:if(e[a]!==t[a])return this2
$.stopEffectAnimation(),void this.startEffectAnimation(t)}o(i,t)),f.highlight=function(){this.trigger("emphasis")},f2
$.downplay=function(){this.trigger("normal")},f.updateData=function(t,e){var i=t.hostModel,this.childAt(0).updateData2
```

之后再搜 startEffectAnimation:

```
$.getShallow("symbolRotate")||0)*Math.PI/180||0;var d={}:if(d.showEffectOn=i.get("showEffectOn"),d.rippleScale=r.get2
$.("rippleEffect.scale"),d.brushType=r.get("rippleEffect.brushType"),/*构造 rippleEffect-num 配置项*/d.num = r.get2
$.("rippleEffect.num"),d.period=1e3*r.get("rippleEffect.period"),d.effectOffset=e/t.count(),d.z=r.getShallow("z")||0,d2
$.zlevel=r.getShallow("zlevel")||0,d.symbolType=a,d.color=l,this.off("mouseover").off("mouseout").off("emphasis").off2
$.("normal"),"render"===d.showEffectOn)this._effectCfg?this.updateEffectAnimation(d):this.startEffectAnimation(d),this2
$. _effectCfg=d;else(this._effectCfg=null,this.stopEffectAnimation():var f=this.childAt(0),p=function(){f.trigger2
$.("emphasis"),"render"!==d.showEffectOn&&this.startEffectAnimation(d)},g=function(){f.trigger("normal"),"render"!==d2
$.showEffectOn&&this.stopEffectAnimation():this.on("mouseover",p,this).on("mouseout",g,this).on("emphasis",p,this).on2
```

大概跟着未压缩的版本对一对，基本就能搞定  
最后，展示一下效果:

num = 1:



num = 2:

