

# 关于Axure中引入Echarts图表的几种方式

## 1. 前言

目前各种数据可视化界面越来越多，在原型设计中也需要加入各种图表效果，而Axure本身没有自带图表元件，因此自能自己重新绘制或者引入第三方元件。常用的有以下几种：

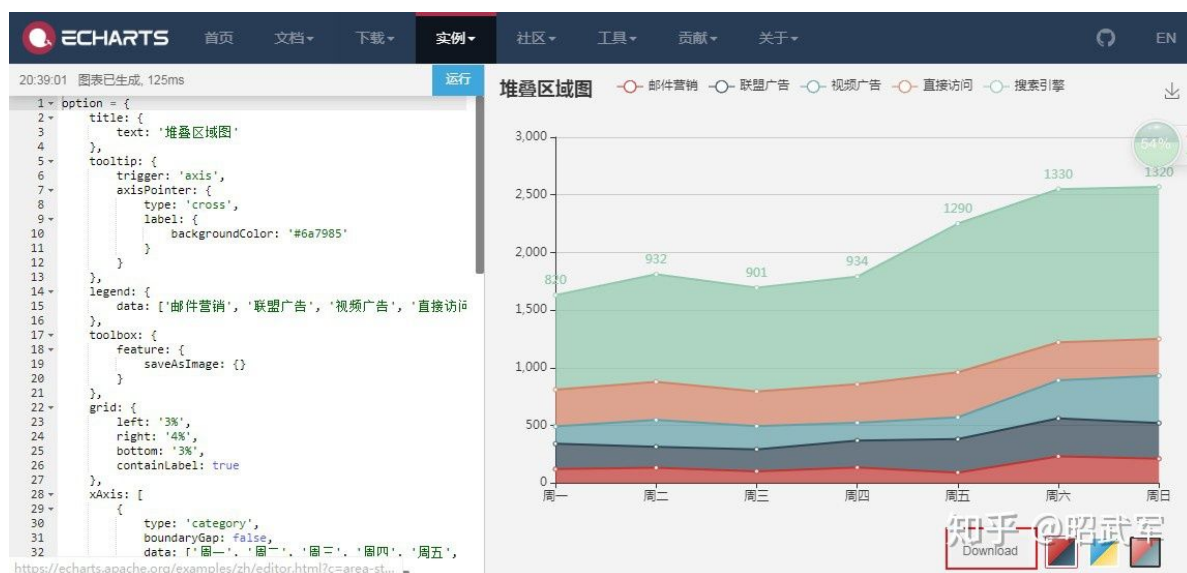
### 1、直接绘制

利用Axure的矩形元件和线条直接绘制，或者通过AI、PS等第三方软件绘制以后截图粘贴过来。这种方式制作比较麻烦，如果图表数量很少，后期改动少的话，可以采用这种方式。

### 2、利用 [EChart](#)、Highchart等图表库在线生成图表，然后通过以下方式引入：

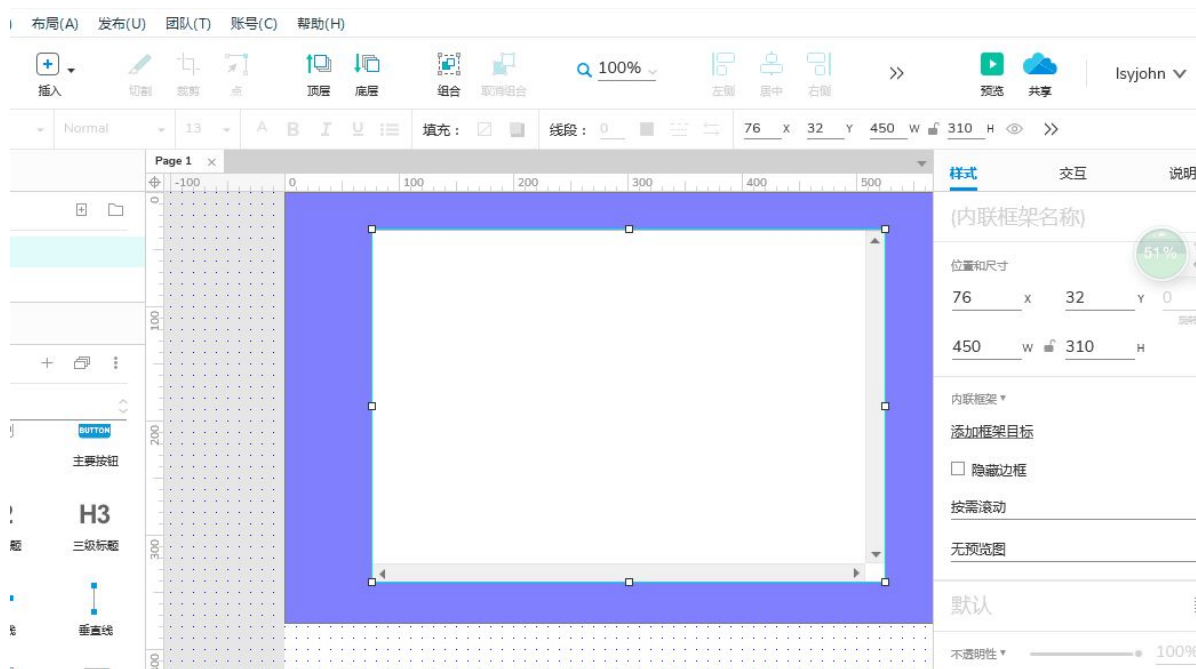
1) 截图引入，将编辑好的图表直接截图到Axure中，方便快捷，一般也能满足要求，但无法保留原Echarts图表的各种交互提示功能；

2) 将 [EChart](#)、Highchart中的实例下载为html页面，然后通过Axure中的内联框架引入。通过这种方式引入，相当于在内联框架所在区域单独引入了一个页面，此时该区域的背景将由引入的页面决定，同时内联框架的大小必须和引入页面相匹配的，不然会出现挤压情况。下面图示为操作步骤，先在 [EChart](#) 上编辑好以后，Download该页面，放在本地文件夹。



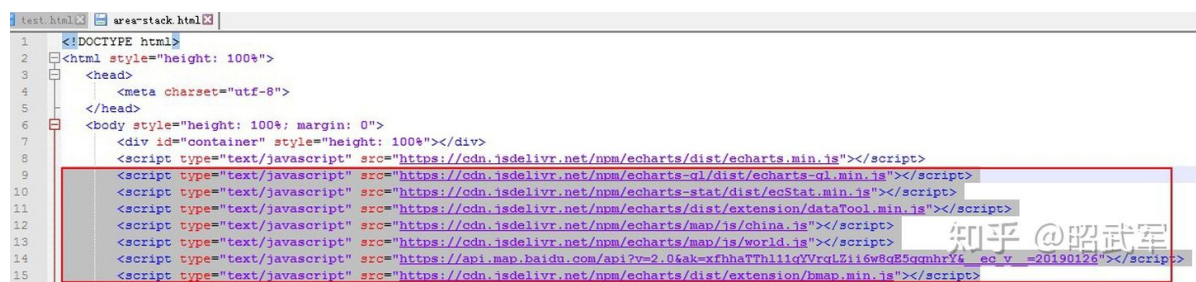
### Echarts下载页面

然后在Axure中插入一个内联框架，链接到文件所在位置即可。该种方式引入时，不能通过预览进行查看，只能生成html页面后才能查看。



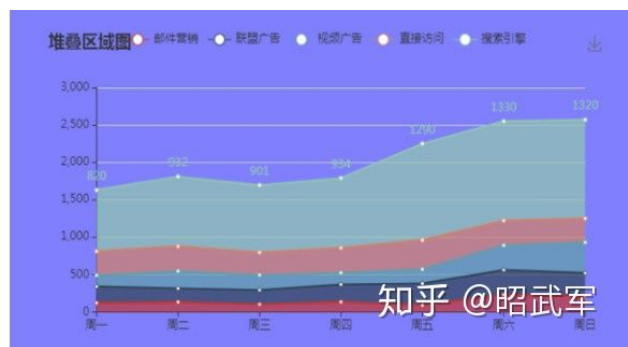
## 插入内联框架

从动图中可以看到，显示时会弹出“百度未授权使用地图API....”的提示，我们用Notepad++打开，把除第一行引用echarts.min.js的其它script引用删除即可。对于大部分没有引用百度底图的图形都可以这样，只保留echarts.min.js。



利用内联框架引入，可以保留Echarts图表原来的交互，但要注意一下两点：

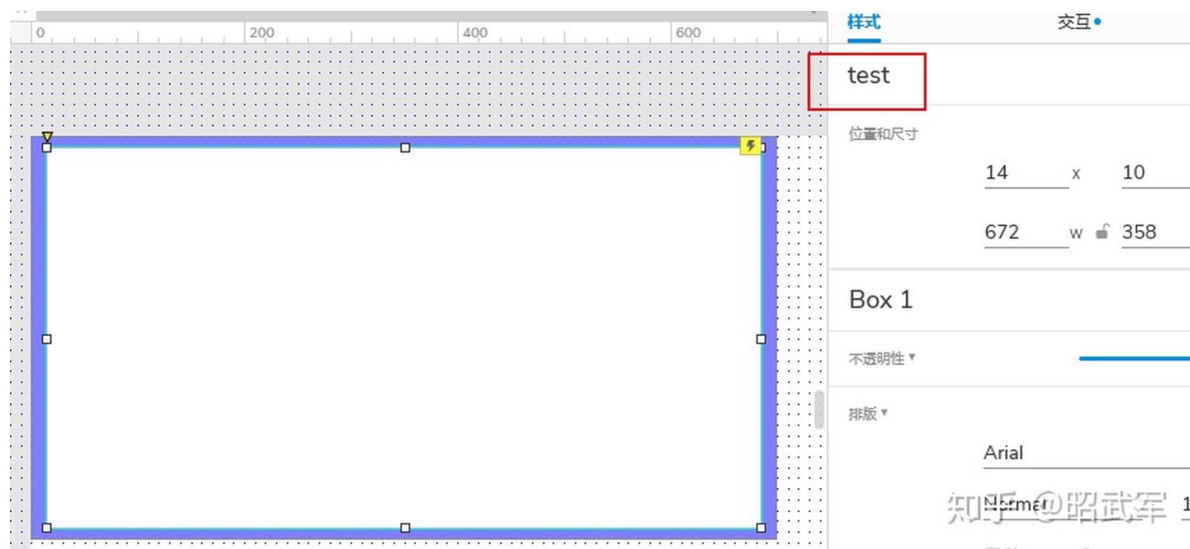
- 1) 内联框架的大小要合适，否则显示会出现挤压；
- 2) 必须生成本地html页面后才能看到引入的图表；
- 3) 图表区域背景无法设置成透明，从下图就可以看出，左边是通过内联框架引入的，右边是通过后面马上介绍的通过加载javascript脚本的方式动态引入Echarts图表。



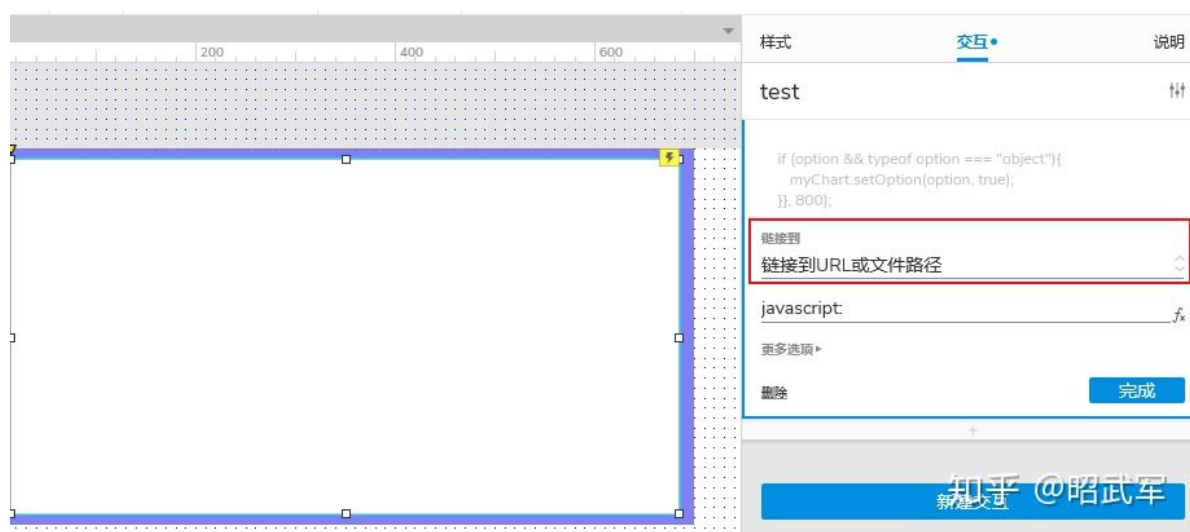
通过javascript伪协议直接执行ECharts、Highchart的代码，动态引入图表

Axure支持通过javascript伪协议执行javascript代码，因此可以引入Echarts的js库，然后直接在Axure中执行图表的绘制代码，具体如下：

1、首先，在Axure中插入一个矩形框，作为Echarts图表绘制的容器，并命名为test，名字可以自定义。为了和上面对比，下面先画了一个矩形框作为背景。



2、然后插入一个载入时事件（如果要单击时才出现，可以插入单击事件），用来触发javascript伪协议的执行。选择“链接到URL或文件路径“，点击fx，在弹出的窗口中可以输入javascript伪协议代码。





3、代码的基础框架如下所示，首先引入Echarts库文件，然后在定时器中执行图标绘制代码，先获取前面插入的矩形框，作为图表绘制的容器，然后进行初始化。只需要将Echarts官网中的示例代码的option部门内容拷贝到代码中的option位置即可。然后将代码整体复制到上面的Axure中的编辑框中。

```
1  javascript:
2  //引入ECharts库
3  var script = document.createElement('script');
4  script.type = "text/javascript";
5  script.src
6  ="https://cdn.jsdelivr.net/npm/echarts/dist/echarts.min.js";
7  document.head.appendChild(script);
8  //获取插入的矩形框对象，test是上面取得名字，如果是其它名字，则进行替换
9  //这里用了定时器来延迟图表载入时间，测试时发现刚引入ECharts库就执行图形
10  //代码可能会出错。
11  setTimeout(function() {
12  //获取插入的test矩形框，作为图表绘制的容器，如果前面矩形框为其它名字，则
13  test改为对应的名字
14  var dom = $('[data-label=test]').get(0);
15  //初始化
16  var myChart = echarts.init(dom);
17
18  var option = {
19  /*在此处粘贴ECharts官网的示例代码*/
```

```

17     };
18     //设置数据
19     if (option && typeof option === "object"){
20         myChart.setOption(option, true);
21     }, 800);

```

代码复制到Axure中时需要去掉其中的所有汉字注释代码，下面为去掉注释的版本。

```

1  javascript:
2  var script = document.createElement('script');
3  script.type = "text/javascript";
4  script.src
5  ="https://cdn.jsdelivr.net/npm/echarts/dist/echarts.min.js";
6  document.head.appendChild(script);
7  setTimeout(function(){
8      var dom = $('[data-label=test]').get(0);
9      var myChart = echarts.init(dom);
10
11      var option = {
12
13      };
14
15      if (option && typeof option === "object"){
16          myChart.setOption(option, true);
17      }, 800);

```

我们将上面的代码拷贝到test矩形框中的编辑框中，然后从EChart中找到想要展示的图表，然后将左边中的option代码全部复制，然后替换掉上面代码中的option={}部分，点击预览即可查看（需要联网）。



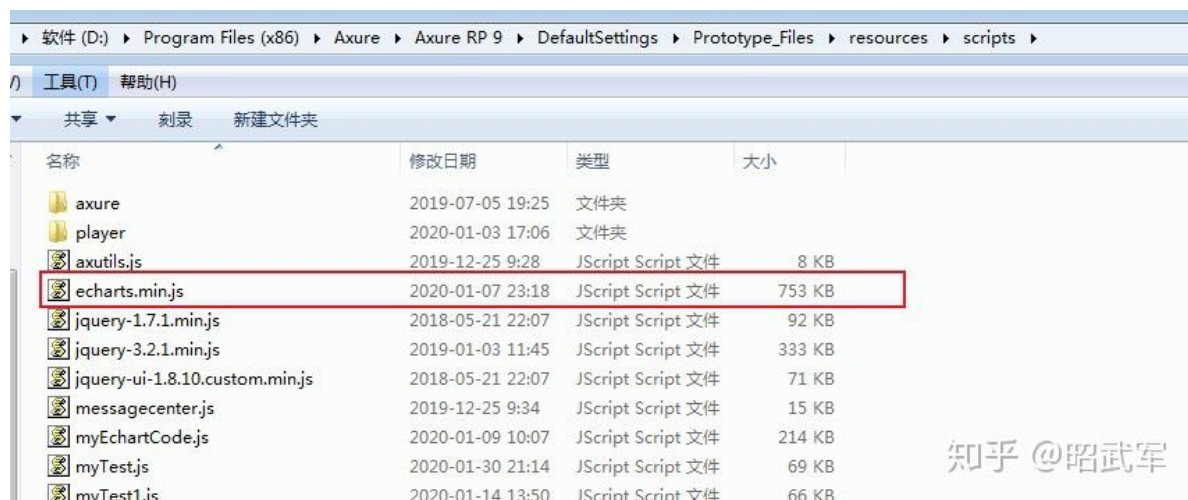
从预览中可以看到，图表本身的背景色为透明色，因此会显示出Axure中设置的背景色。通过该方式引入的图表可以实时预览查看，并可以显示Axure中设置的背景色，保留Echarts图表本身的交互动画，但是需要联网。



## 1.1 脱机使用方法

如果脱机使用的话按一下方法：

把官网下载的js文件放在Axure的安装目录下，具体为D:\Program Files (x86)\Axure\Axure RP 9\DefaultSettings\Prototype\_Files\resources目录下，只要在这个目录下就可以正常引用，推荐放在该目录下的scripts文件夹中。



然后在引用代码中改为`script.src = "resources/scripts/echarts.min.js"`即可。这样在生成html源文件的时候，该js文件会一同被复制到目标目录中，这样我们直接运行html文件也可以正常显示。

## 1.2 最后

对于Hightchart或者其它的javascript图标库均可采用类似的方法进行引入，借助Echarts官网中提供的表格工具，我们可以很方便的将excel中的数据转为代码可用的数组，从而可以更方便的使用Echart进行数据的展示。



