前端学习

AVUE-DATA

数据大屏解决方案

一个很多骚操作的前端框架

目　录

项目介绍
项目启动
    前端启动
    Jave版后端启动
    Node版后端启动
基础操作
    1.新建空白大屏
    2.画布介绍
    3.添加组件
    4.调整组件图层位置
    5.预览、保存组件
基础组件
    1.背景配置
    2.图表类组件
        2.1柱形图组件
        2.2折线图组件
        2.3饼图组件
        2.4环形图组件
        2.5象形图组件
        2.6雷达图组件
        2.7散点图组件
        2.8漏斗图组件
    3.文本类组件
        3.1文本框组件
        3.2跑马灯组件
        3.3超链接组件
        3.4实时时间组件
    4.图片类组件
        4.1图片组件
        4.2图片框组件
        4.3轮播图组件
        4.4滑动组件
        4.5iframe组件
        4.6video组件
    5.指标类组件
        5.1翻牌器组件
        5.2仪盘表组件
        5.3字符云组件
        5.5进度条组件

6.表格类组件
6.1表格组件
6.2选项卡组件
7地图类组件
7.1地图组件
8万能组件
8.1堆叠条形图
8.2正负条形图
8.3双向对比柱形图
8.4圆形柱形图
8.5嵌套饼图
8.6矩形树图
8.7k线图
9万能dataV组件
自定义Vue组件
全局变量
组件数据交互
组件参数交互
组件联动交互
二次开发
外部引入
内部引入
项目部署
Nginx部署
1.项目打包
2.Linux部署
3.Baota部署
3.1安装宝塔
3.2基础部署
3.3大屏部署
引入其它项目使用

项目介绍

数据大屏解决方案

介绍

Avue-Data数据大屏(Vue全家桶+ElementUi+Echart+dataV)开发，丰富的交互控件和图表组件，提供智能图形推荐，报表图形任意切换，且不受维度,度量的限制。

特点

- 缩短开发周期

  提供丰富的二次开发接口
- 快速响应

  支持动态局部刷新，秒级响应
- 多端部署

  适配各种拼接大屏场景
- 实时数据

  支持多种数据源接入数据

多种数据源

- 在线API
- 静态数据
- SQL数据

20+常用组件

- 图表
  - Echart通用型
  - DataV
  - 柱状图
  - 折线图
  - 饼图
  - 象形图
  - 雷达图
  - 散点图
  - 漏斗图
  - 地图

- 文字
  - 文本框
  - 跑马灯
  - 超链接
  - 实时时间
- 媒体
  - 图片
  - 图片框
  - 轮播图
  - iframe
  - hls播放器
  - video播放器
- 指标
  - 翻牌器
  - 环形图
  - 进度条
  - 仪表盘
  - 字符云
- 表格
  - 选项卡
  - 表格
- 边框以及各种常用组件，同时支持自定义组件

其他大屏例子

- API接口
- SQL在线
- 组件互动
- 动态参数交互
- 选项卡例子
- 柱状图和折线图
- Echart通用配置

丰富模板

政府行业

> BI大数据可以打破信息壁垒，形成统一数据共享大平台，打破“信息孤岛”问题。

教育行业

BI大数据可以有效查看各基地的情况，通过对教、学、研多层面的数据整合，制定更行之有效的方案。

金融行业

BI大屏拥有收集、运用数据的能力，从而做出快速精准的应对策略。

销售行业

BI大屏可以通过信息技术将各个渠道的数据进行整合，更好的对客户信息进行挖掘、创造价值。

交通行业

BI大屏在整个交通运输系统中的应用能有效解决城市化加速而导致的各种问题，在大客流运营常态下有着重要的价值。


智能交通综合执法平台-执法管理
首页
案件办理
联合惩戒
执法监管
执法系统分析
社会公示和服务
待办事项
预警审核 (80条)
立案登记 (22条)
登记审核 (6条)
处理意见 (4条)
案件移送 (35条)
违法通知 (19条)
处罚决定 (0条)
结案归档 (12条)
预警数 1425件
立案数 5305件
结案数 3300件
出具文书合计 2560件
违法行为类型统计
单位（件）
本月
上个月
超限超载
抽烟
不按路线行驶
违停
违规上下客
打瞌睡
疲劳驾驶
行政处罚程序统计
简易程序 4860件
一般程序 1370件
听证程序 2450件
30%
45%
25%
近3个月违法车辆类型统计
违法最多车辆类型：公交车
违法最少车辆类型：网约车
货车
客车
公交车
出租车
网约车
次数
时间：2019.04.10-2019.07.10
执法单位信息
立警为公 执法为民
单位名称：深圳市南山区粤海街道交通管理局
所属管辖范围：南山区粤海街道
机构：交通管理执法机构
科室：交通执法科室
人数：23人
通知公告
1、严格纪律作风，强制提升交通行政执法队伍形象；
2、加强执法队伍班子建设，不断增强班子的凝聚力、向心力和战斗力；
3、加强基层基础建设，夯实交通执法工作根基；
4、强化培训兵，提升执法人员业务素质。
案件办理结果统计
单位（件）
行政处罚数
责令改正数
不予处罚数
案件移送数
强制执行数
8月
9月
10月
11月
12月
1月
2月
3月
4月
5月
6月
7月
今年与去年环比分析
+35%
+14%
-12%
违法数
立案数
结案数

项目启动

项目启动

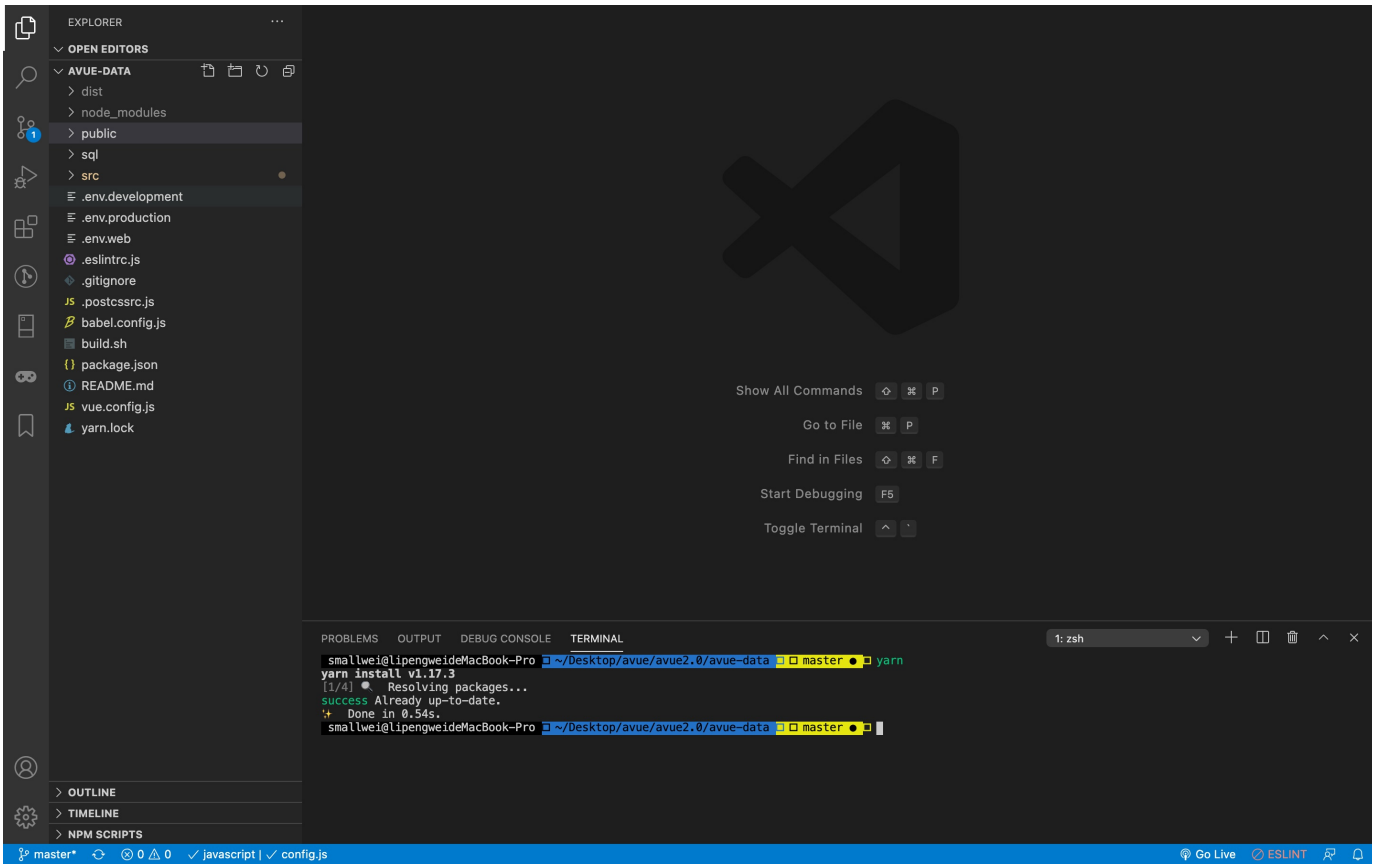
> 需要启动前端和后端2个端才可以正常运行

前端启动

一、工程导入

1. 使用Avue商业账号登录git私服：https://git.avuejs.com/avue/avue-data

2. 导入工程,在终端执行yarn install或者npm install

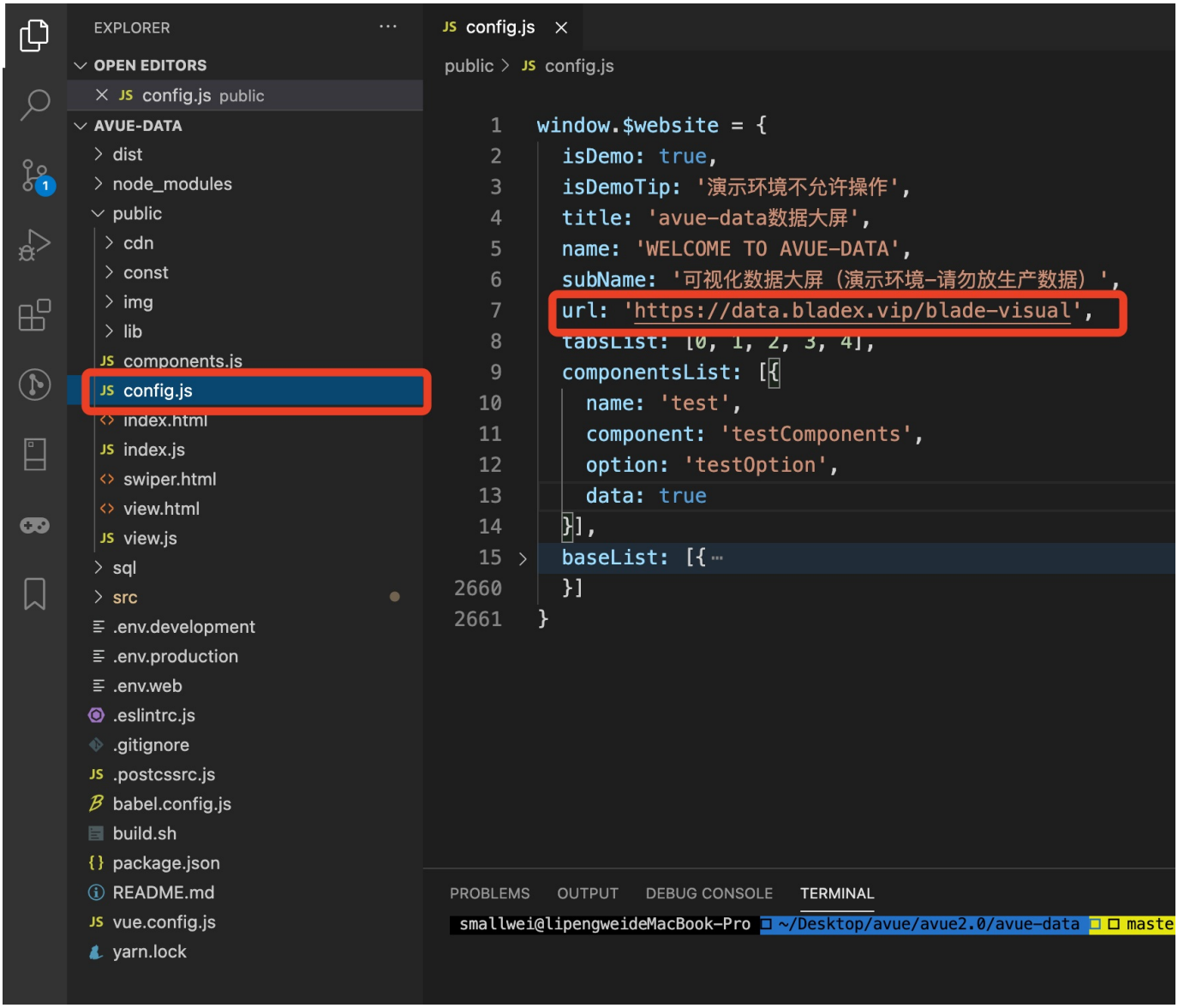


二、配置修改

1.这里要运行大屏后端

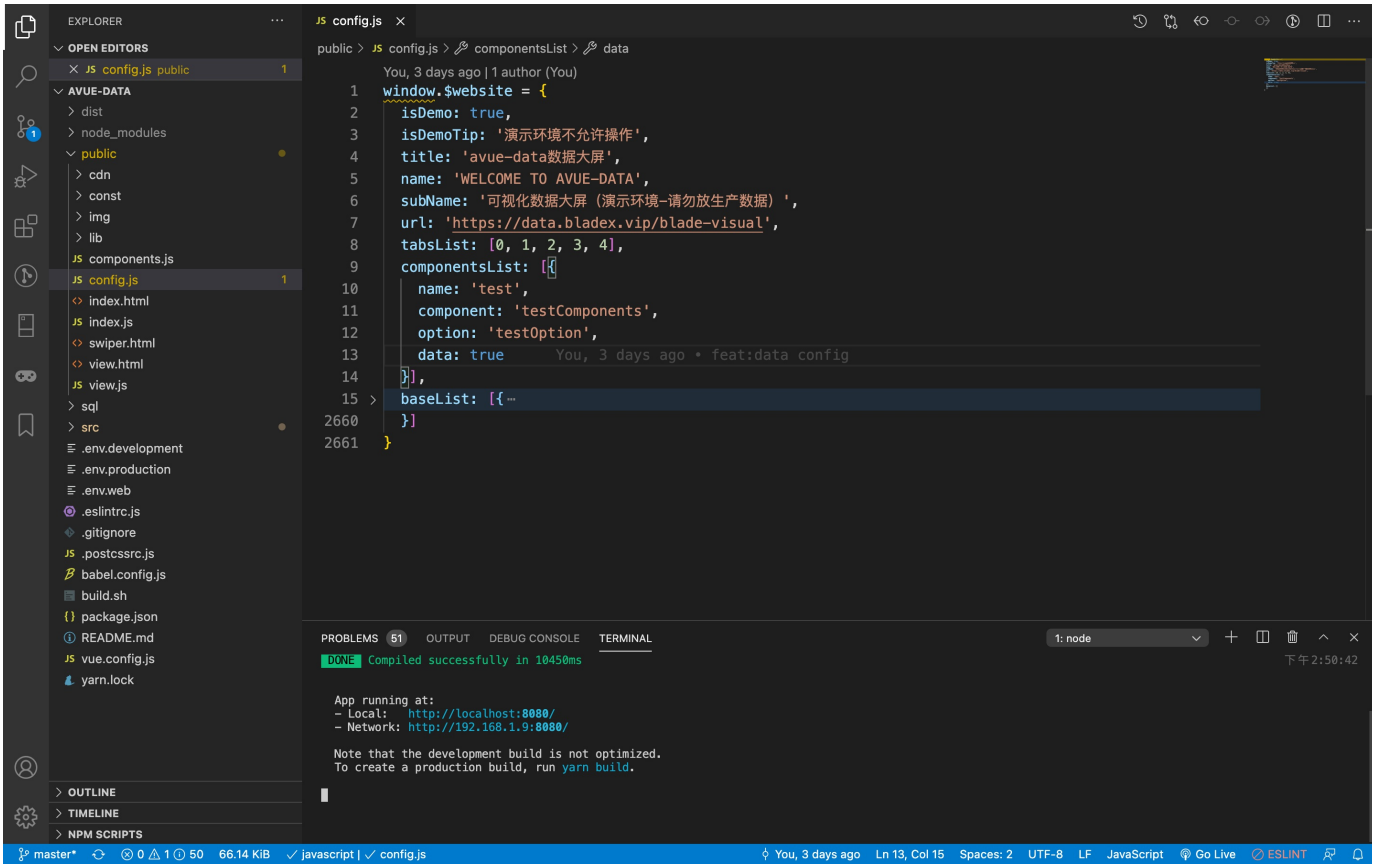
- JAVA版后端启动
- NODE版后端启动

2. 修改你自己的后端地址，默认是演示线上接口



三、工程启动

1. 命令行执行 `yarn run serve` ，看到如下日志则说明启动成功



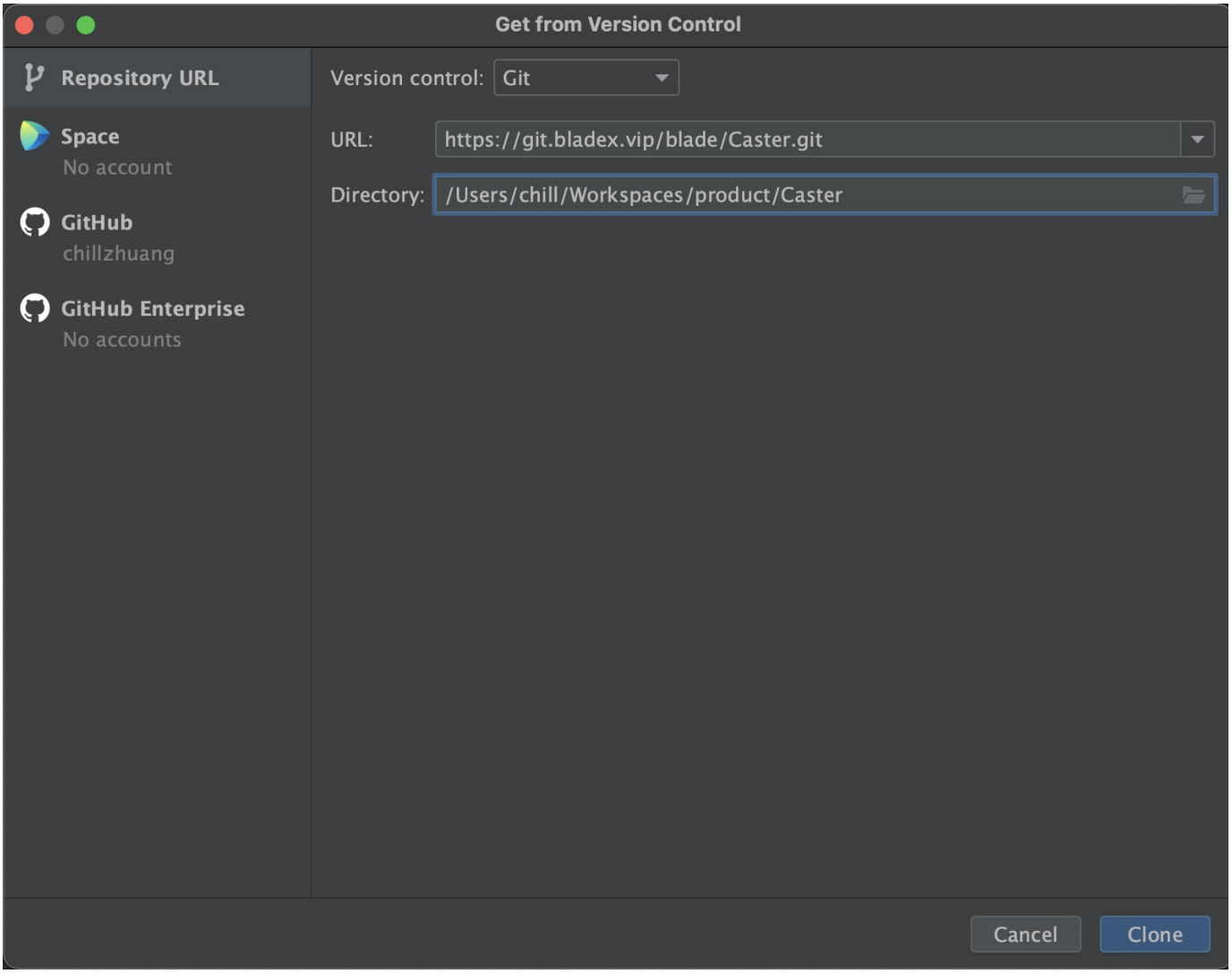
2. 访问 http://localhost:8080 查看效果

Jave版后端启动

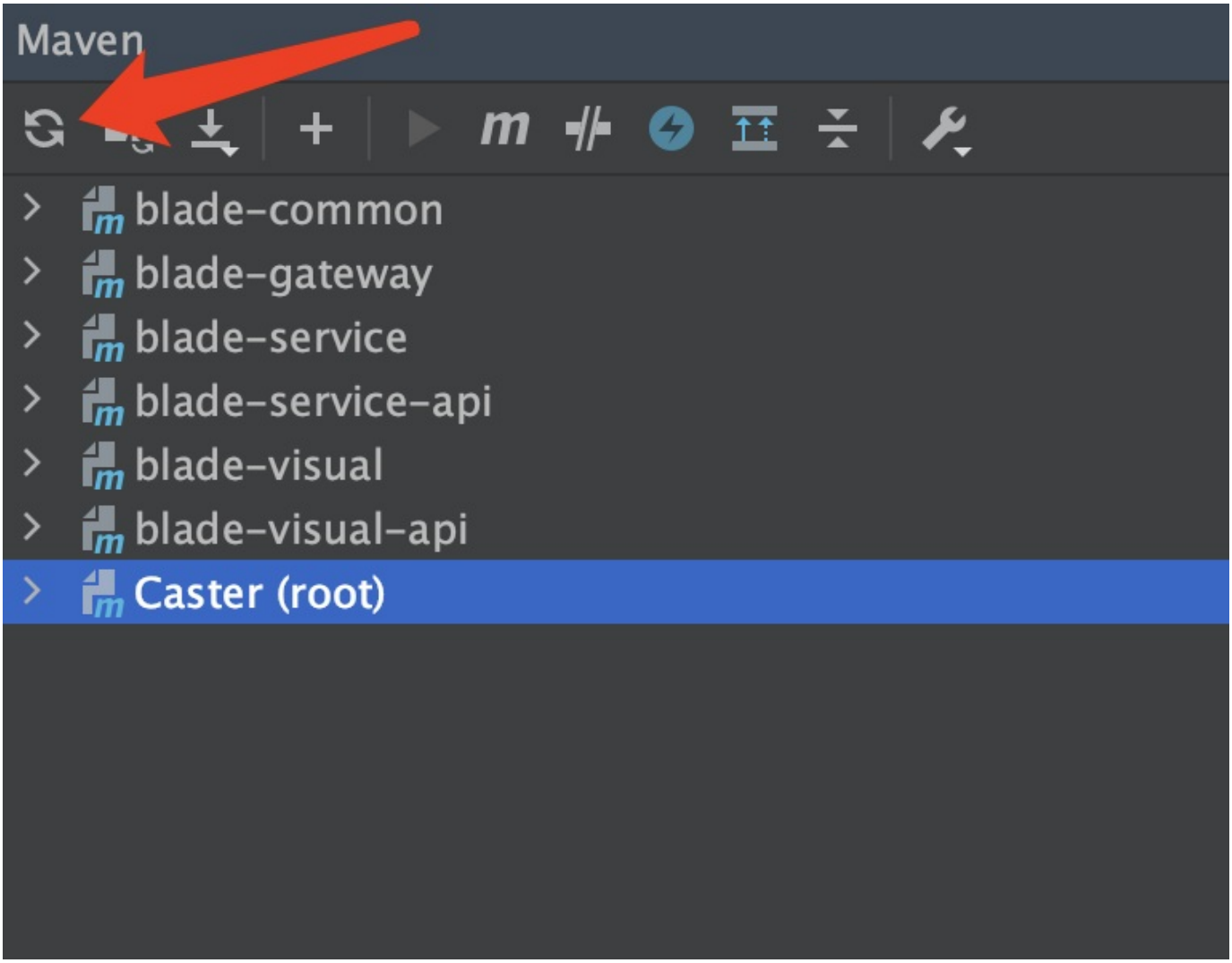
一、工程导入

1. 使用BladeX商业账号登录git私服：https://git.bladex.vip/blade/Caster
2. 复制地址

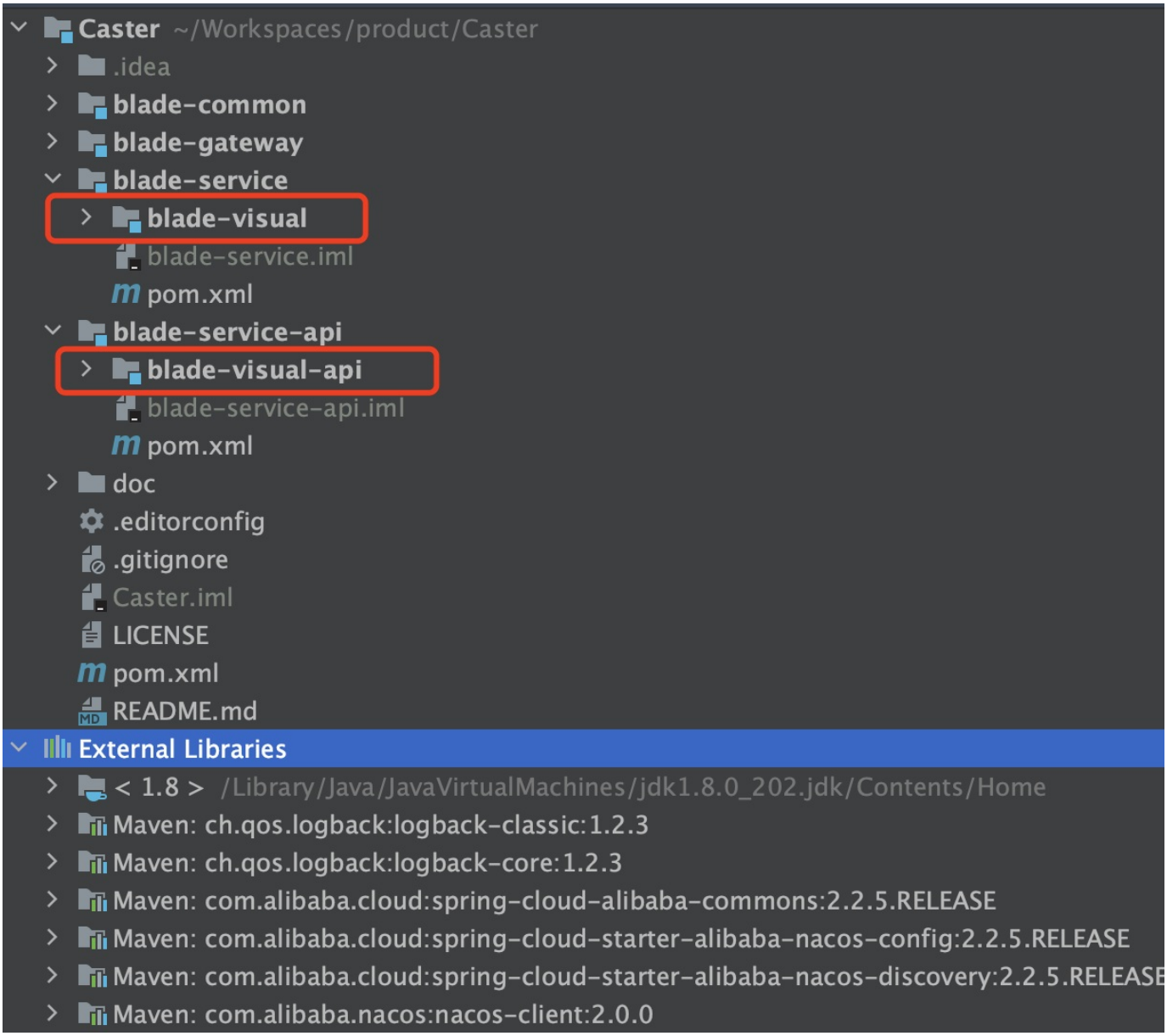
3. 导入工程



4. 导入成功后等待maven依赖加载完毕，如果依赖下载失败请多点击刷新按钮

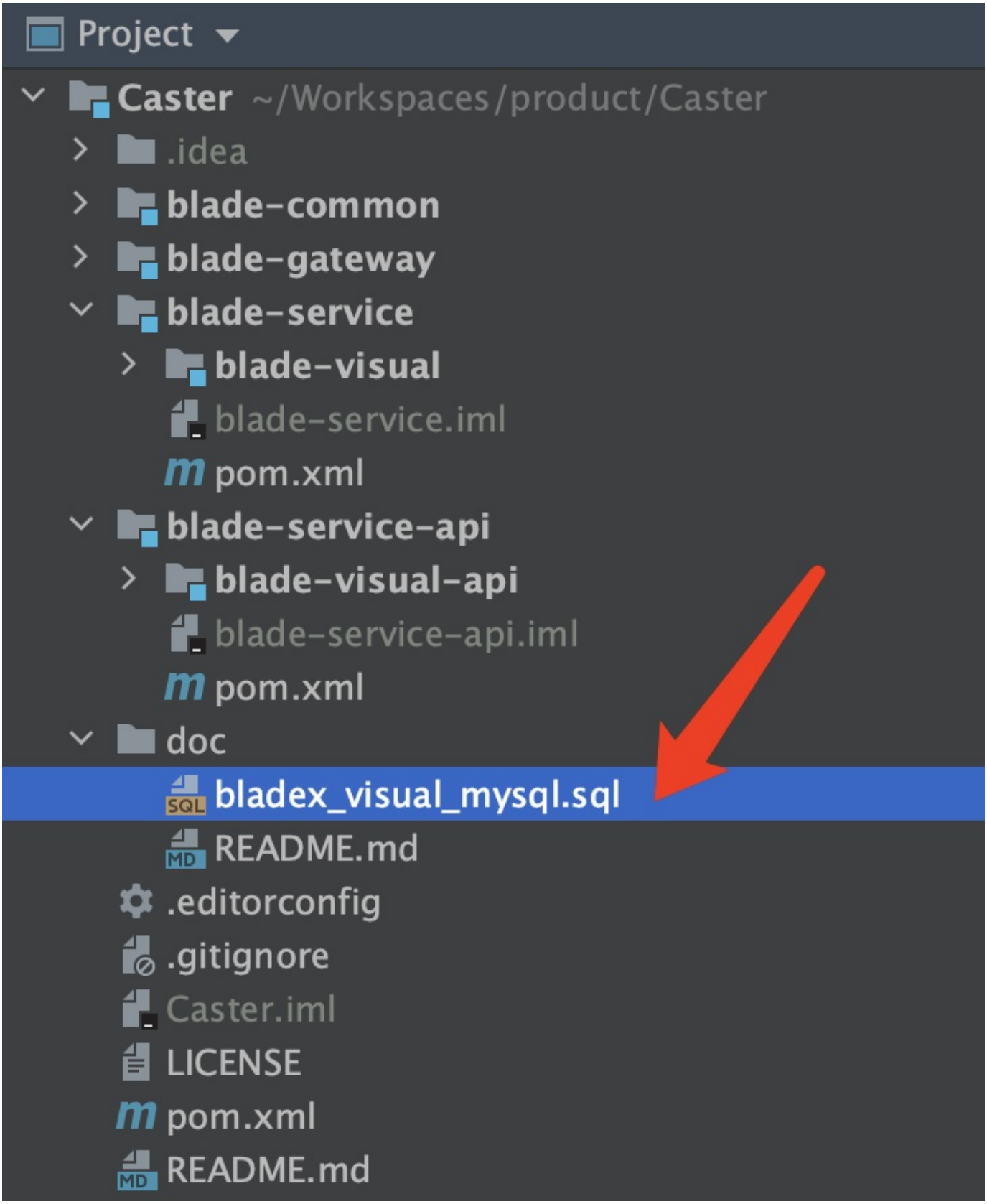


5. 工程构建完毕后，便可以看到核心服务了

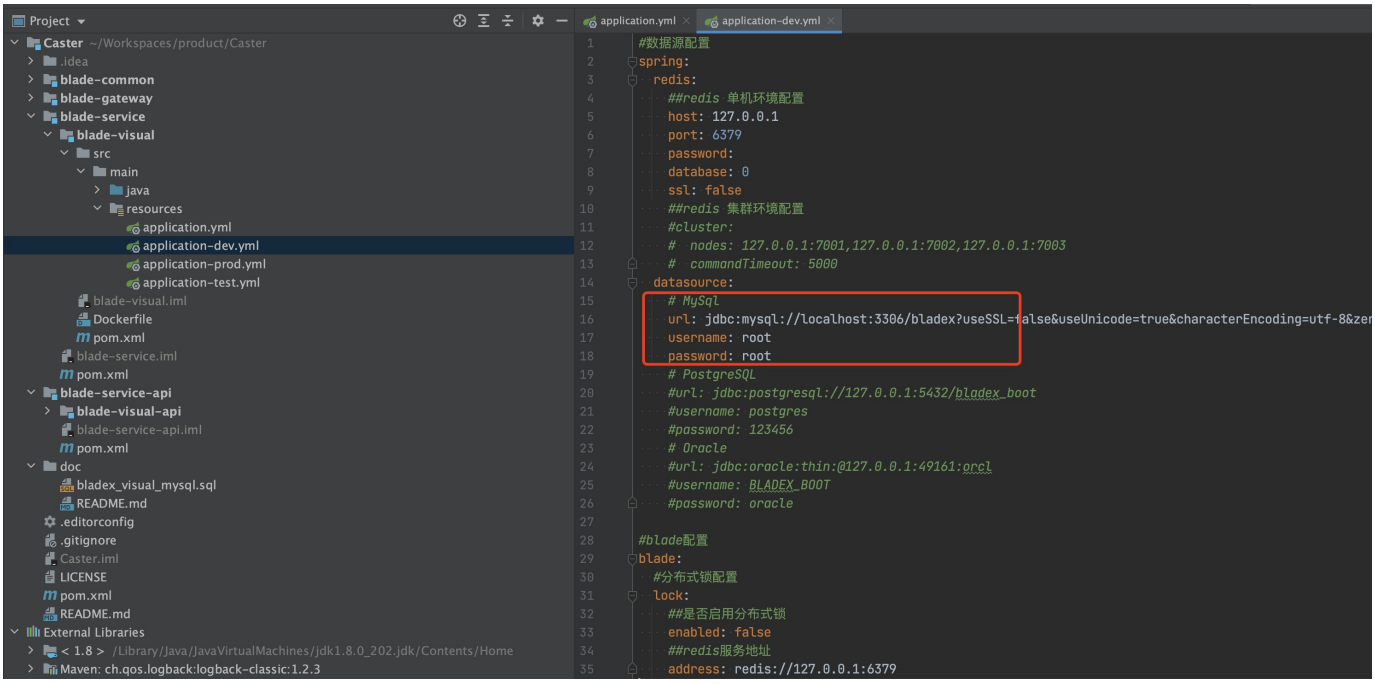


二、数据库配置

1. 创建一个空白数据库并导入如下sql

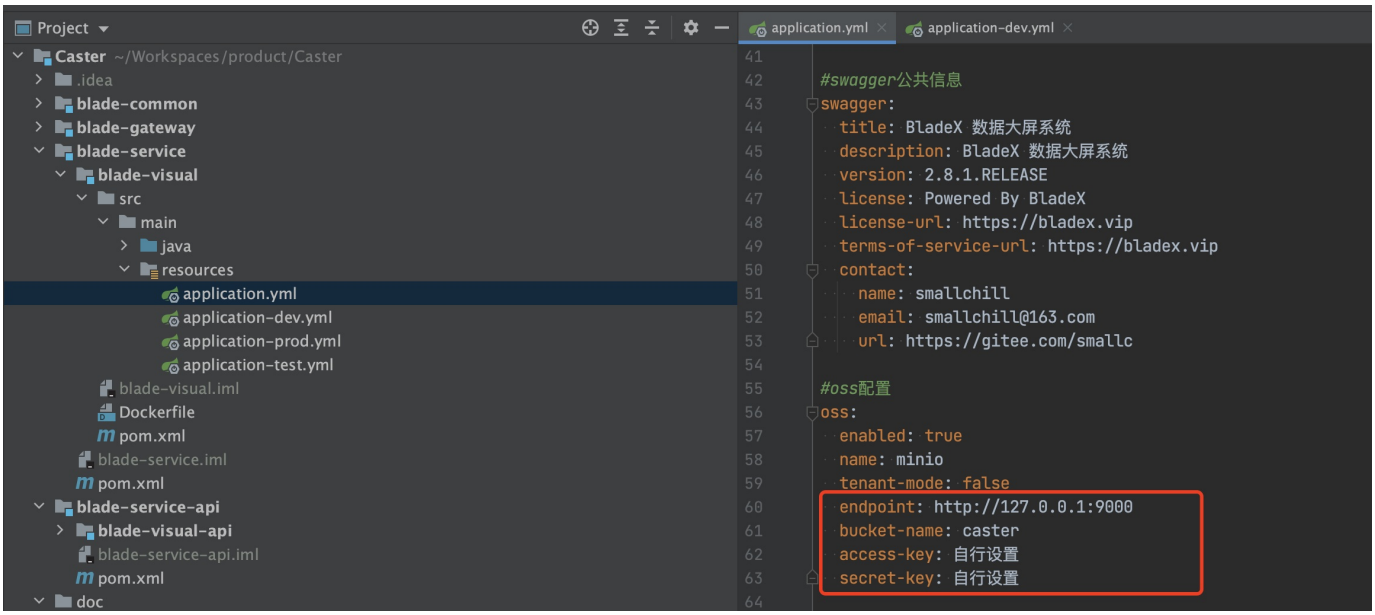


2. 到对应配置文件将数据库信息配置准确



三、对象存储配置

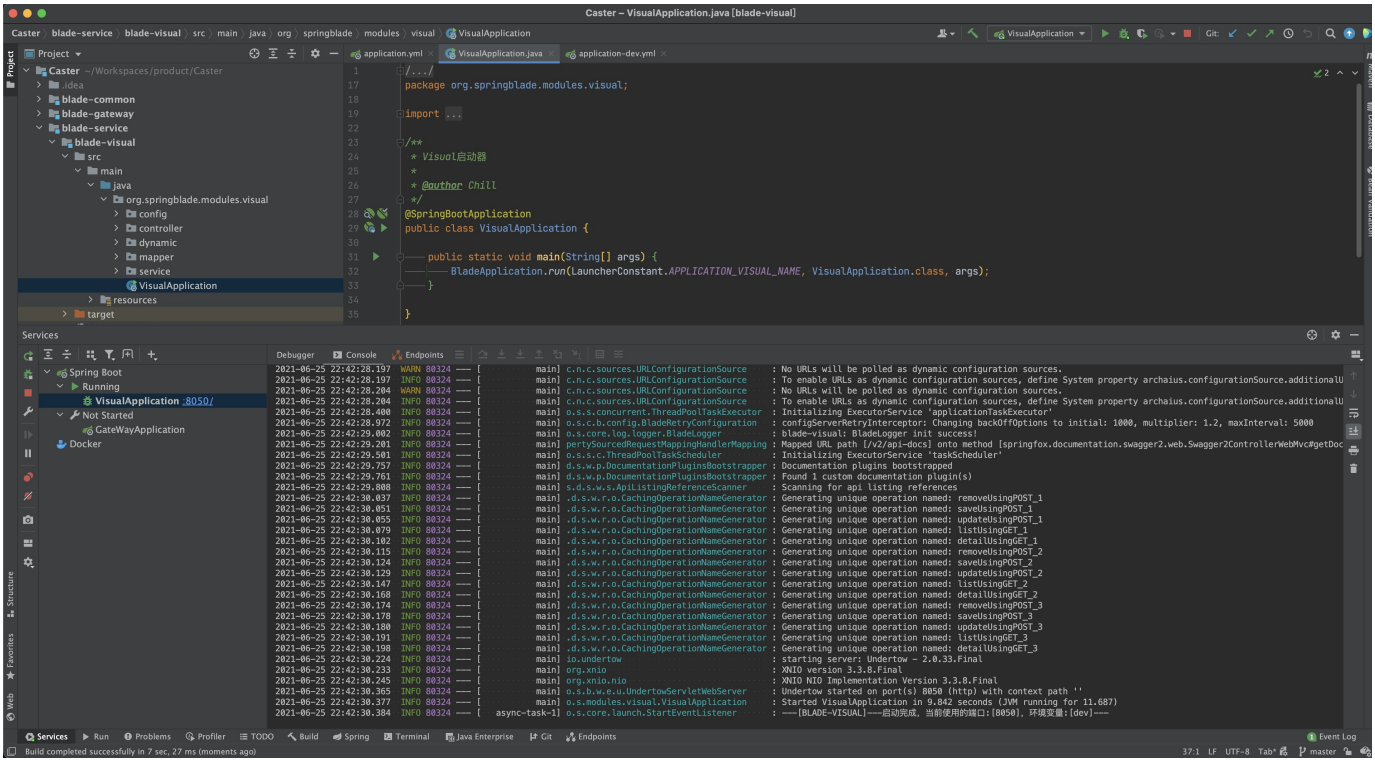
1. 框架默认为minio，如何启动部署请参考官网文档：http://docs.minio.org.cn/docs/
2. 获取minio的相关配置后，将其配置到application.yml文件



3. 若需要配置其他的对象存储，详情请见 `BladeX开发手册 5.13章节`

四、工程启动

1. 一切准备完毕，我们打开VisualApplication，右键启动
2. 若配置无误，我们就可以看到如下日志



五、后记

- blade-visual工程默认为springboot版本，直接启动便可，不需要对接nacos等中间件
- 若需要将其对接至cloud架构也非常简单，具体请见如下步骤：
-

1. blade-visual目录下的pom.xml修改依赖 将

```
<dependency>
<groupId>org.springblade</groupId>
<artifactId>blade-core-boot</artifactId>
<exclusions>
    <exclusion>
        <groupId>org.springblade</groupId>
        <artifactId>blade-core-cloud</artifactId>
    </exclusion>
</exclusions>
</dependency>
```

修改为

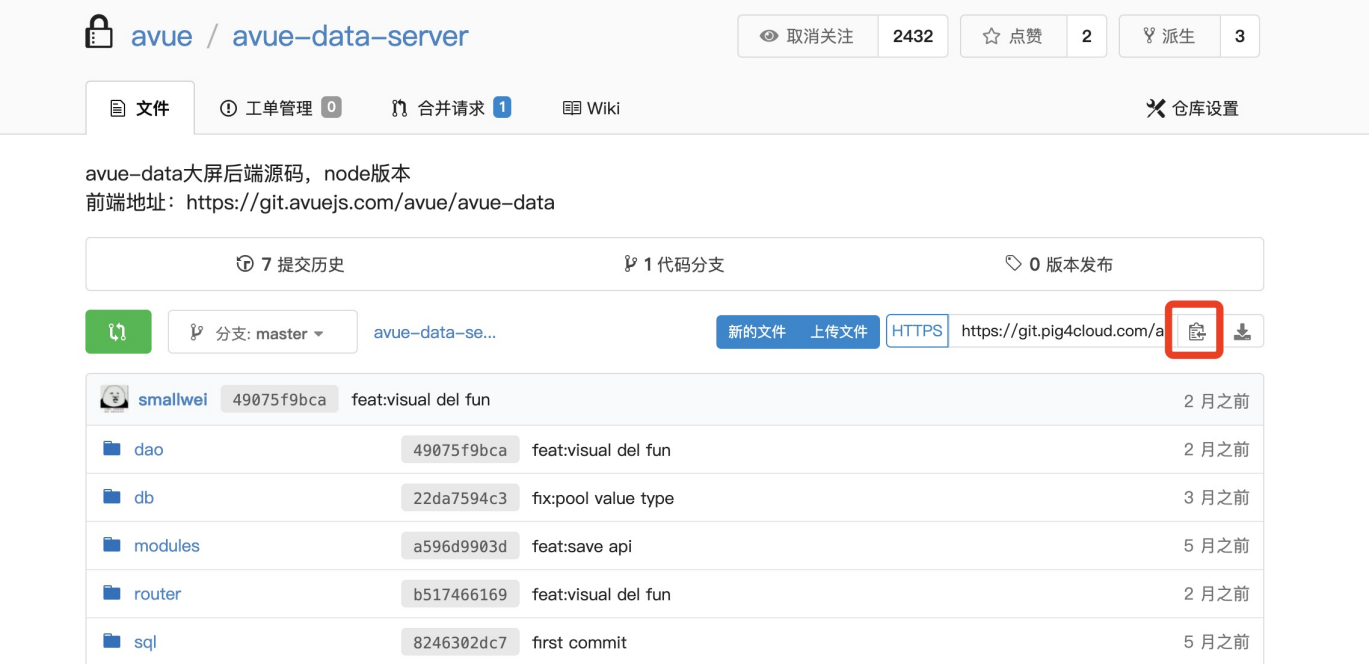
```
<dependency>
<groupId>org.springblade</groupId>
<artifactId>blade-core-boot</artifactId>
</dependency>
```

2. 修改 `VisualApplication` 头部的注解，将 `@SpringBootApplication` 修改为 `@SpringCloudApplication`
3. 修改controller的key前缀，将控制器头部的 `LauncherConstant.APPLICATION_VISUAL_NAME` 删除
4. 修改application-xx.yml，将配置从nacos读取
5. 以上步骤做完后启动 `VisualApplication` 即可对接至cloud
6. 为了方便管理，也可以将修改完毕后的 `blade-visual` 工程完整拷贝至 `BladeX` 的 `blade-ops` 目录下

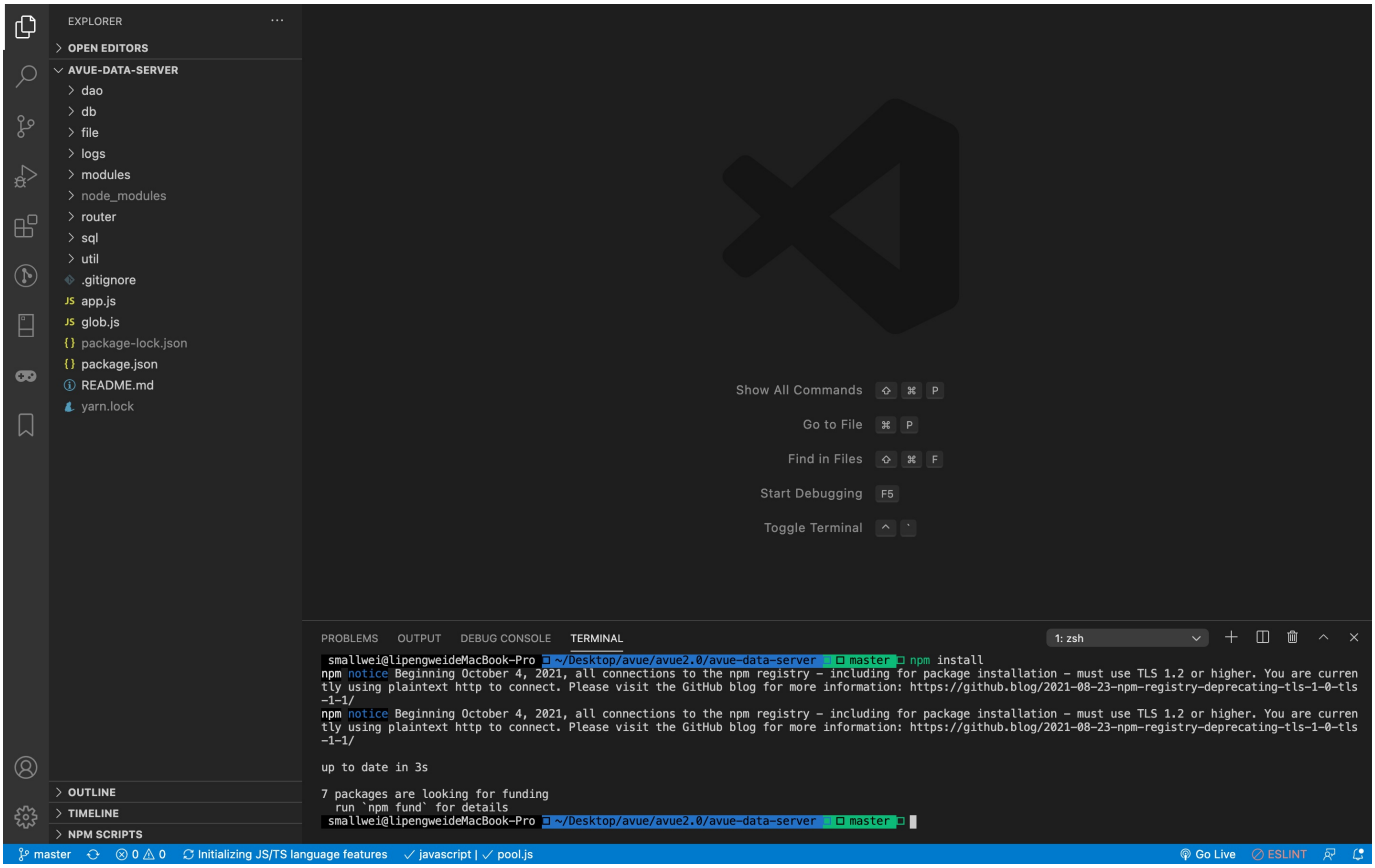
Node版后端启动

一、工程导入

1. 使用Avue商业账号登录git私服：https://git.avuejs.com/avue/avue-data-server

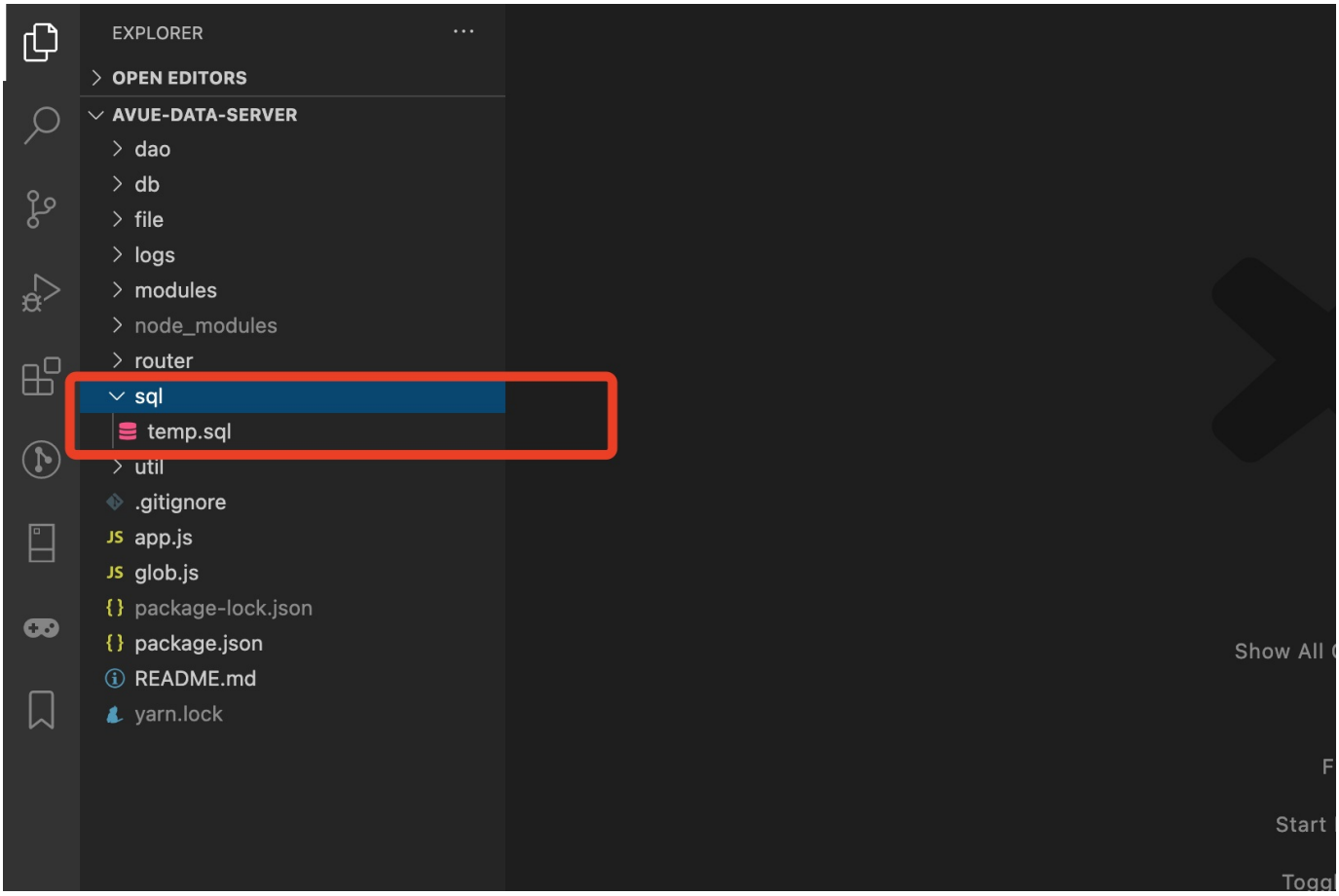


2. 导入工程,在终端执行yarn install或者npm install



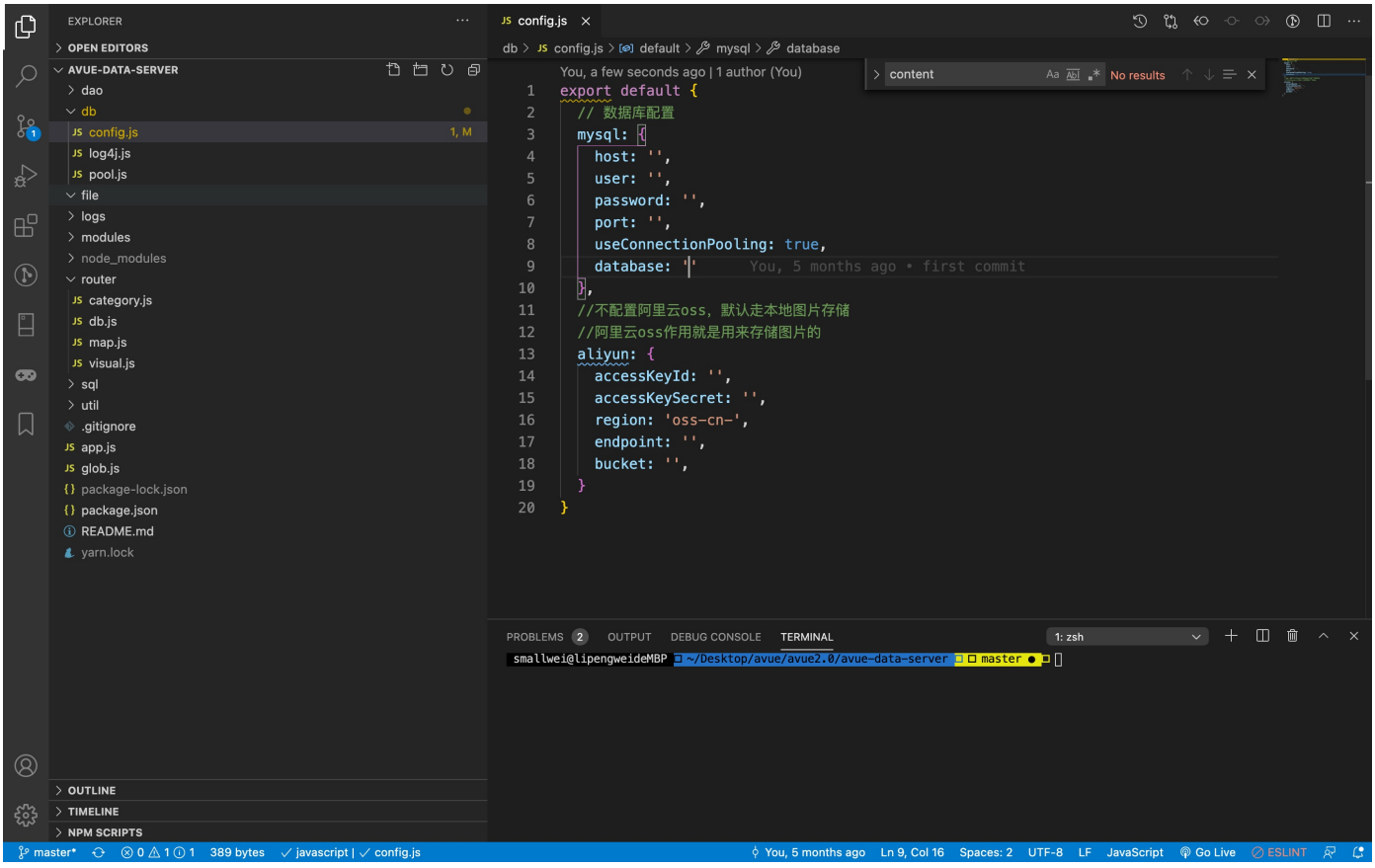
二、配置修改

1. 导入Sql数据文件



2. 修改数据库的连接地址和阿里云oss(阿里云oss作用就是用来存储图片)

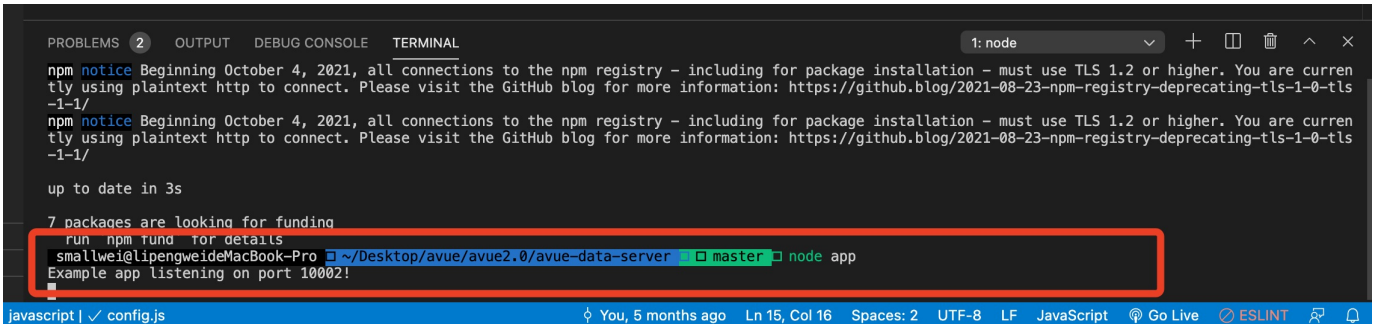
说明:不配置阿里云oss，默认走本地图片存储



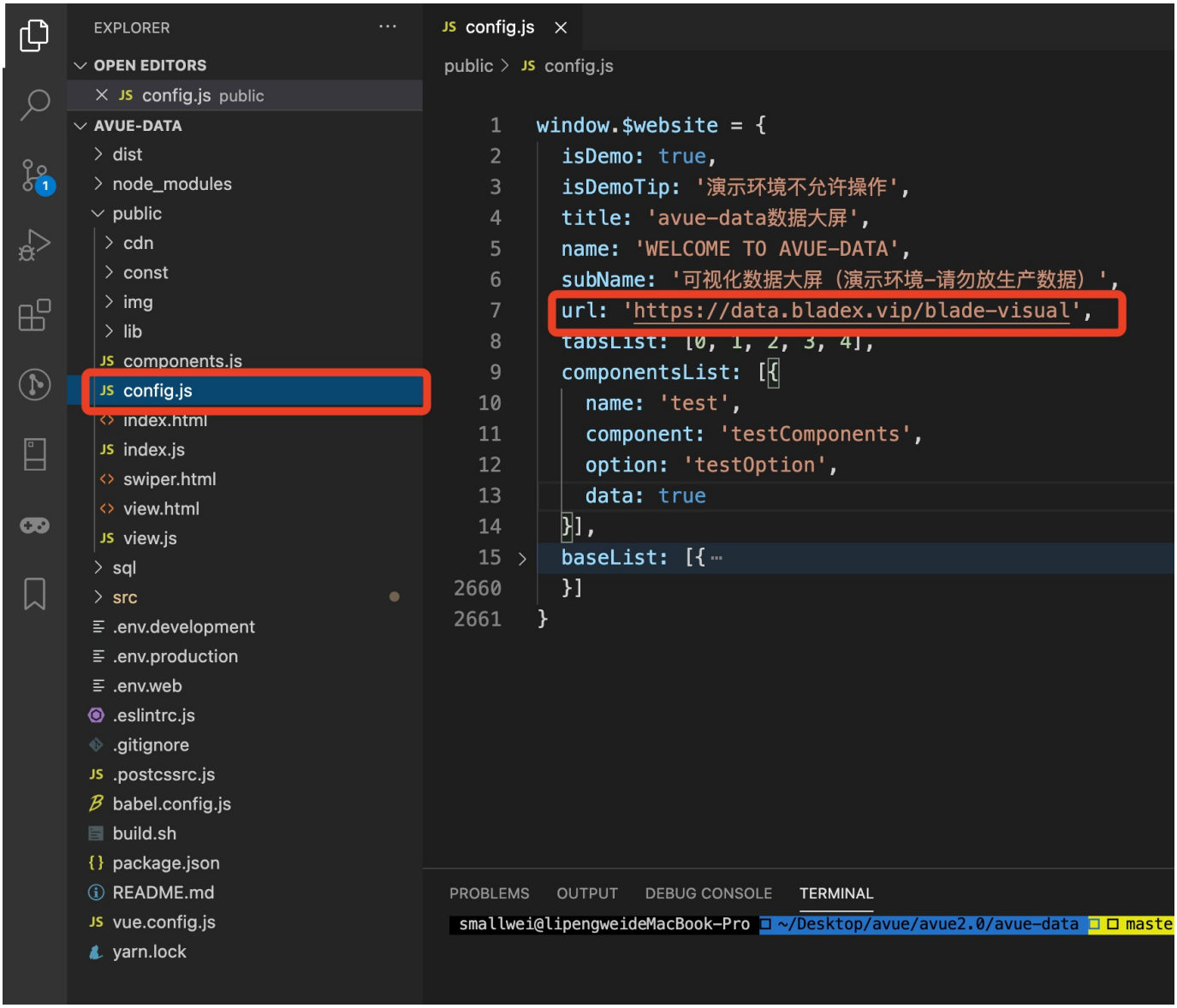
> !

三、工程启动

1. 命令行执行 `node app` ，看到如下日志则说明启动成功



2. 前端项目中修改成改地址这里是修改前端项目



3. 访问前端http://localhost:8080 查看效果


WELCOME TO AVUE-DATA
可视化数据大屏（演示环境-请勿放生产数据）
大屏管理
地图管理
分类管理
数据源管理
综合功能
选择下面的方式进行创建
精选模板
组件库
测试4
测试6
wasu
新建大屏
共 4994 条
50条/页
1
2
3
4
5
6
100
前往
1
页
数据大屏模板
已发布

基础操作

1.新建空白大屏

2.画布介绍

3.添加组件

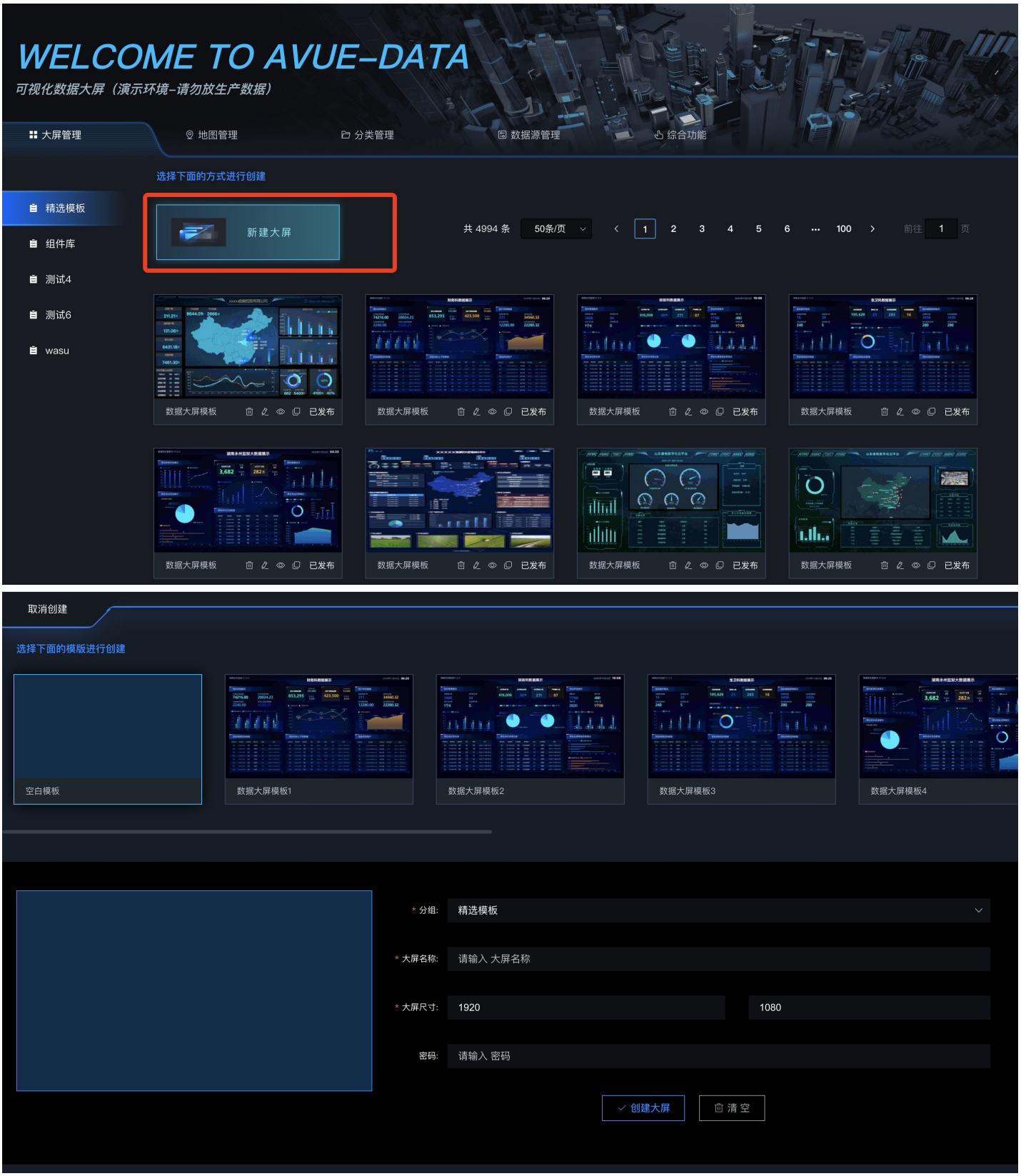
4.调整组件图层位置

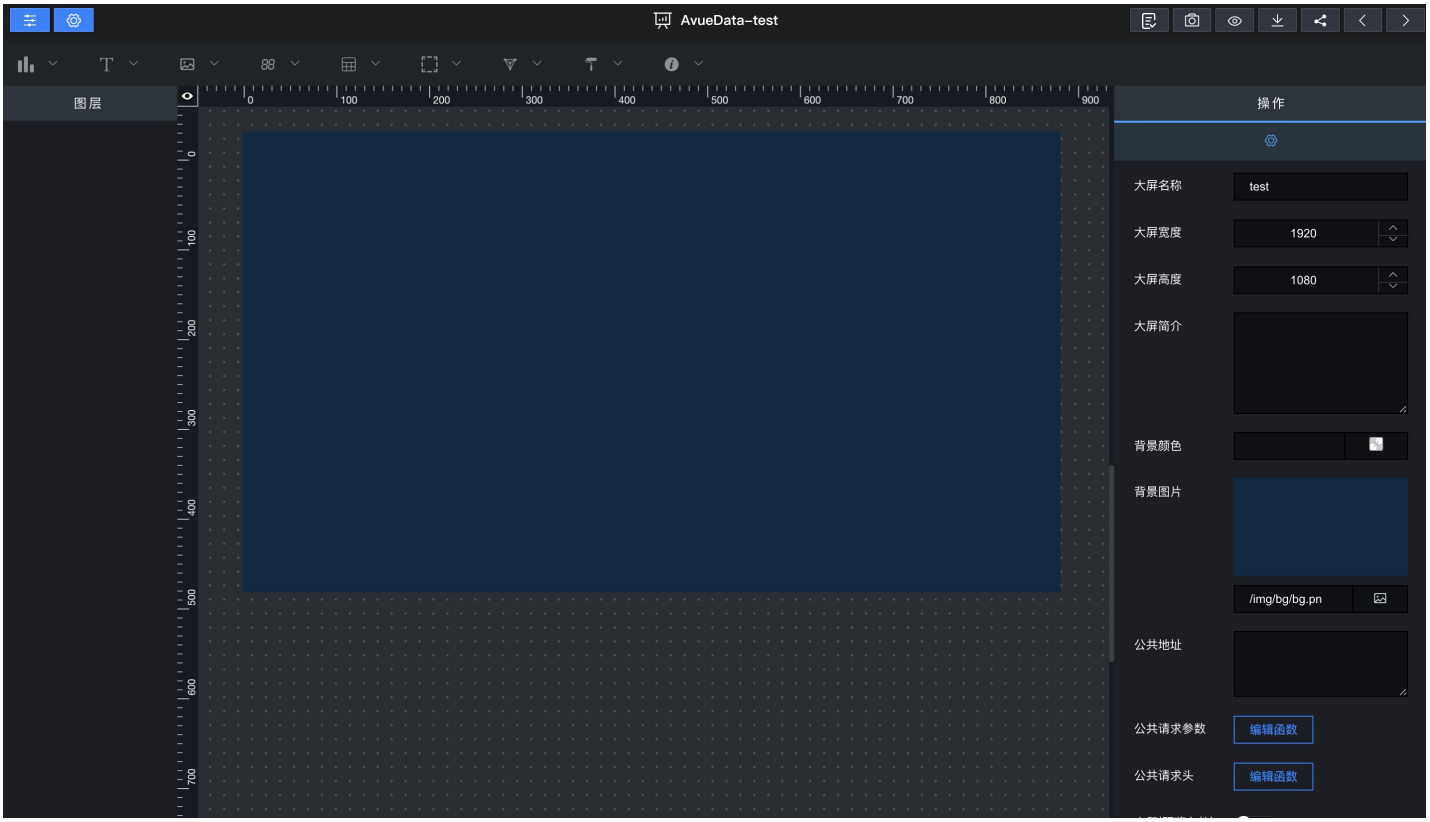
5.预览、保存组件

1.新建空白大屏

一、新建大屏

> 进入大屏设计界面后，点击图中新建大屏”按钮，填写相关信息，即可新建空白大屏；





二、使用模板新建大屏

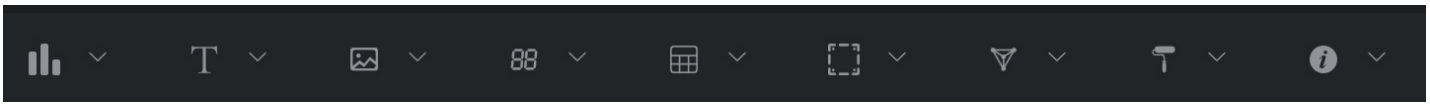
- 如果你想设计的大屏跟模板的类似，在新建的时候没有，可以点击模板上的复制按钮，复制一个新的大屏，在上边修改就可以

- 新建或者复制模版新建大屏的时候，需要设置密码，避免别人修改，或者删除

2.画布介绍

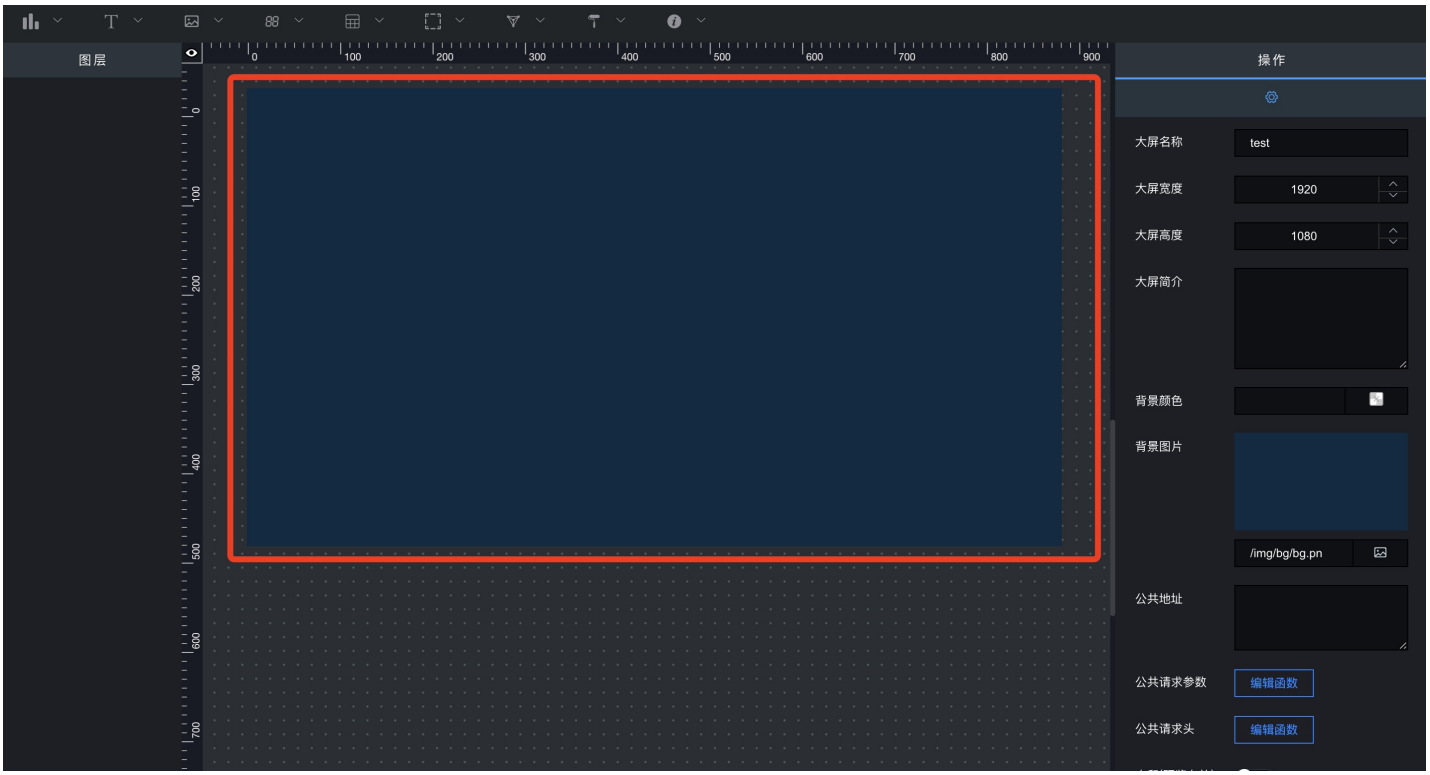
一、组件栏

> 画布顶部为组件栏，可以点击使用任何组件



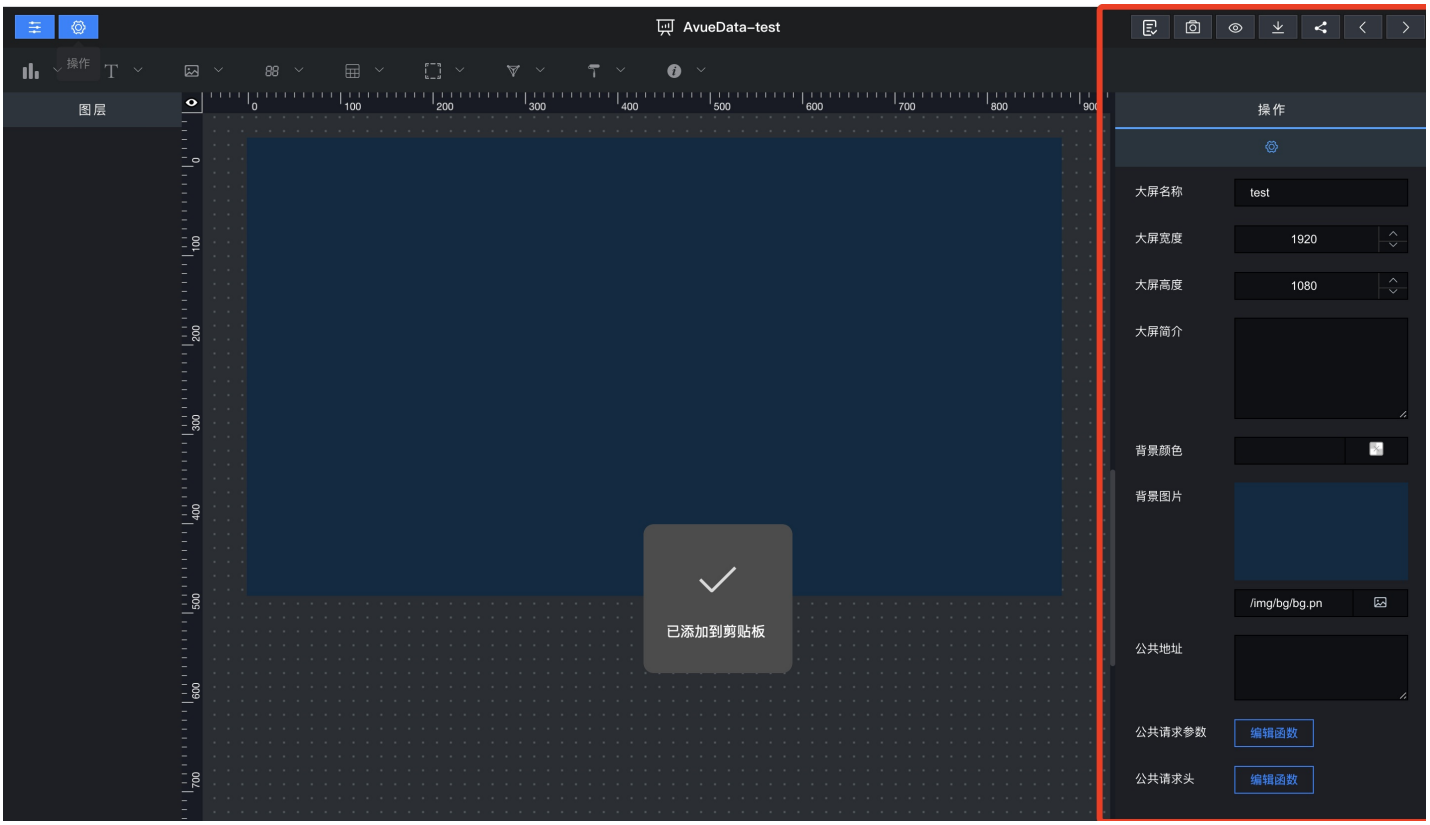
二、画布区域

> 标注的地方为画布区域，可以在画布内使用任何组件进行设计；



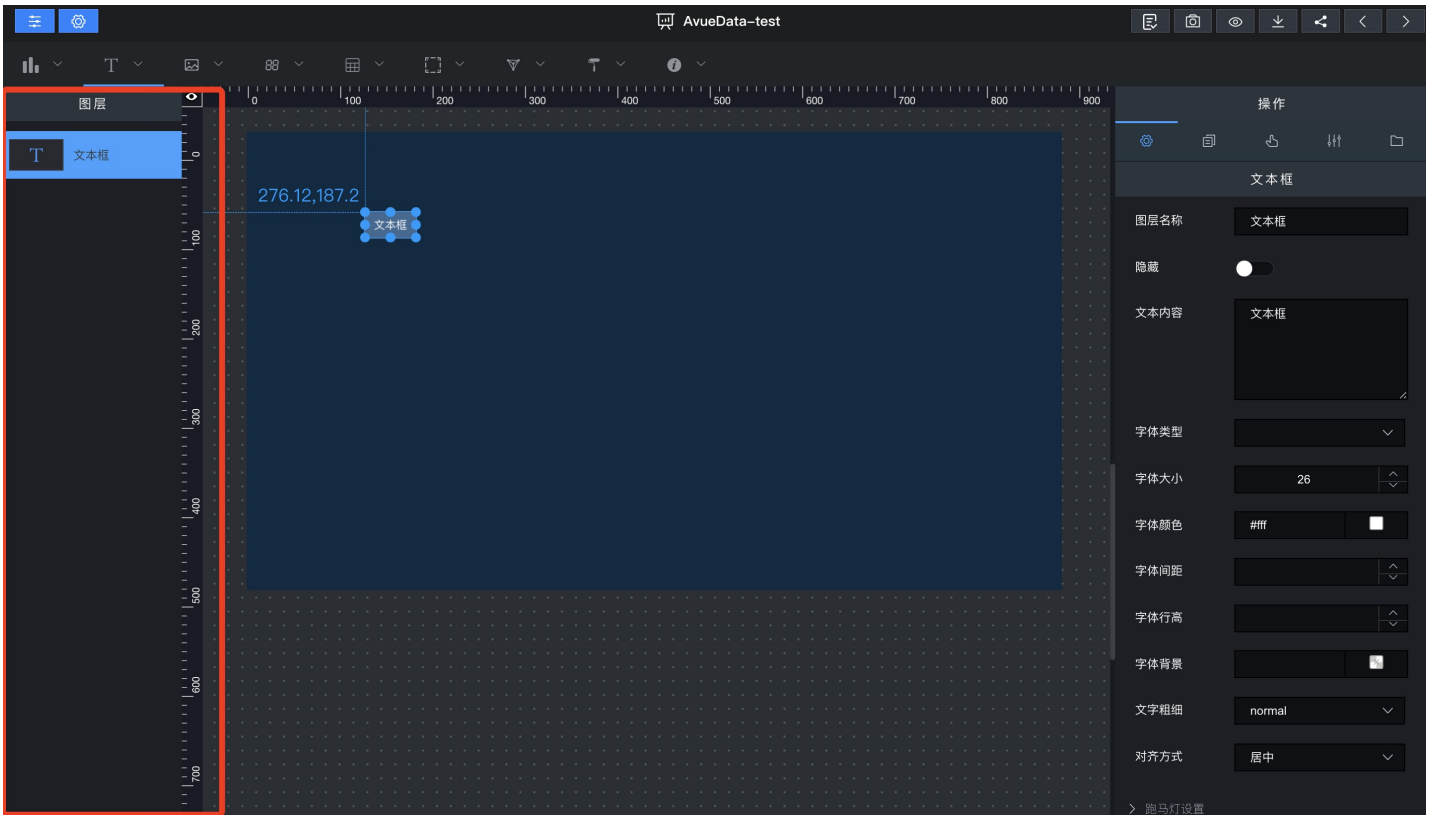
三、样式区域

> 画布右侧为样式区域可以通过调整样式区域的参数来设置画布的大小、颜色等；



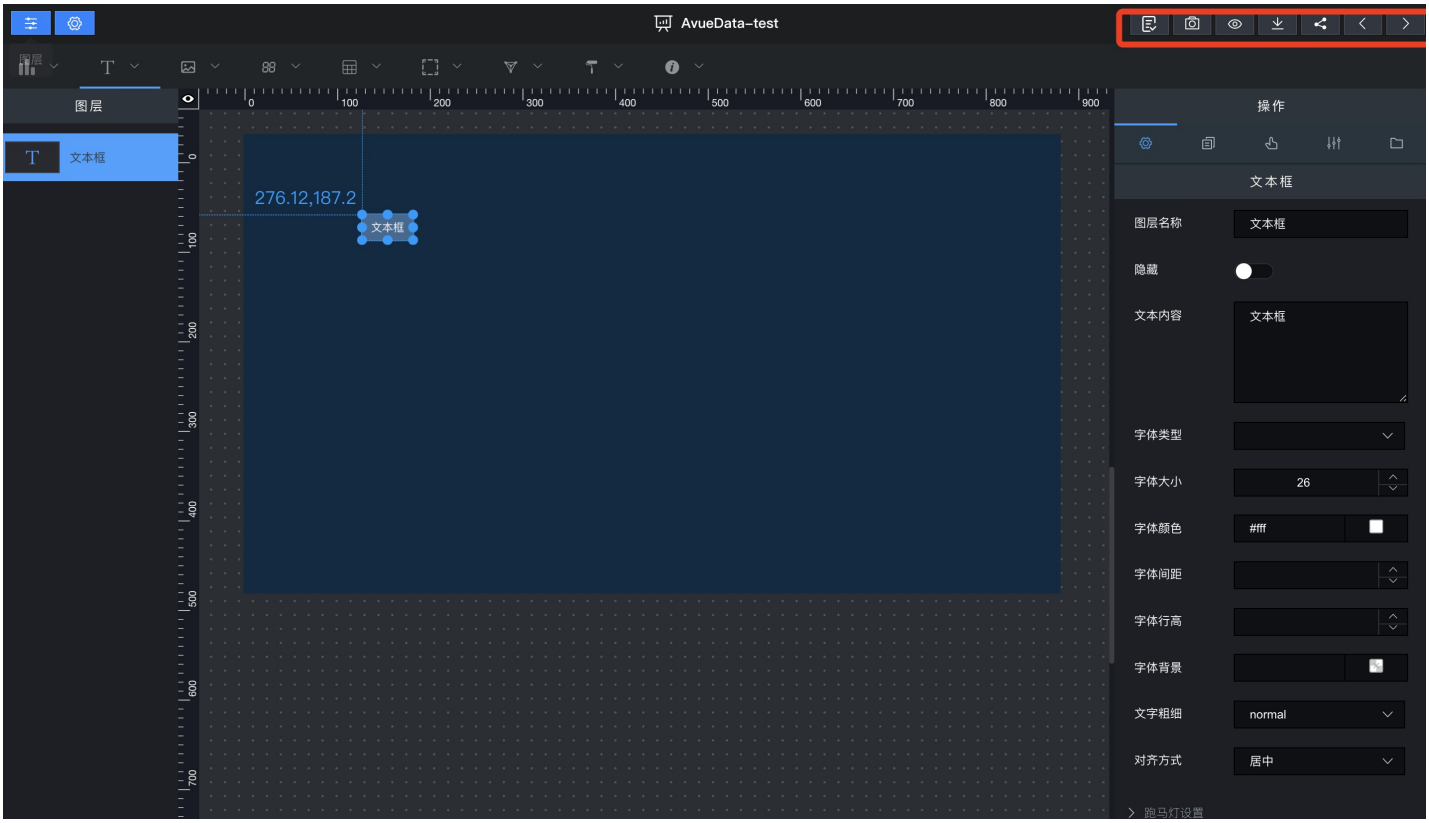
四、图层的区域

> 画布左侧为图层区域，如图3.14；可以通过名称快速定位到画布中的组件；



五、操作栏

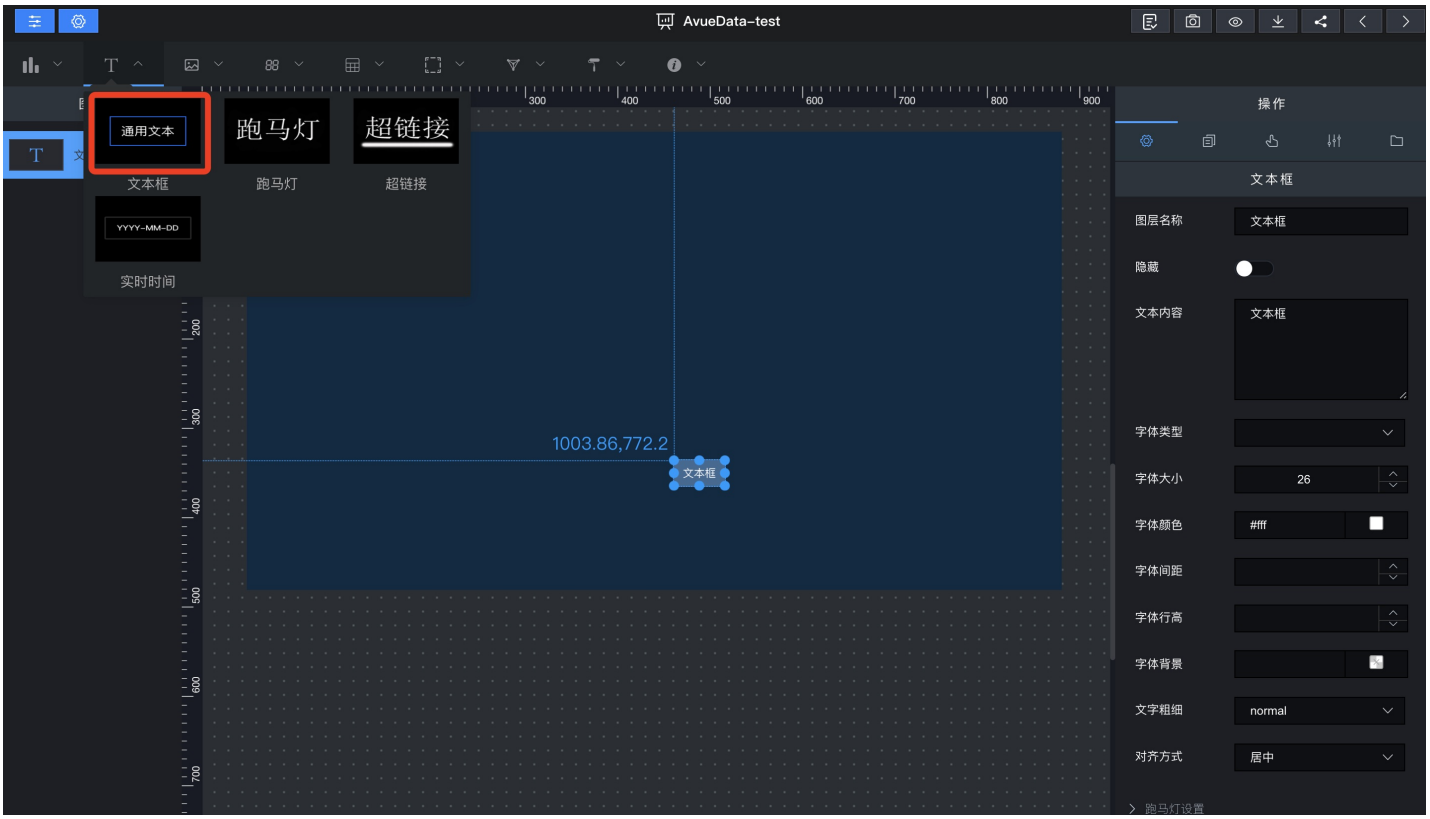
> 可以对大屏进行保存，导出，回退等相关操作



3.添加组件

添加组件

> 在导航栏中，点击要的组件，即可完成组件的添加



4.调整组件图层位置

一、拖动调整

> 可以通过鼠标拖动，改变图层位置；也可以通过托、拽组件，使组件变大或缩小；

二、样式板块调整

> 点击图层右侧画布样式板块中的

”或“

，就可看到调整图层位置的设置

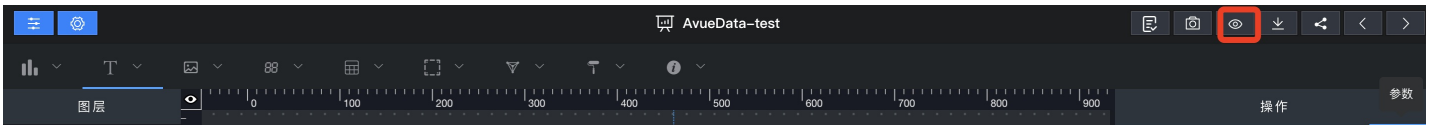
- 宽度：图层的宽度；
- 高度：图层的高度；
- X位置：图层距离X轴的位置；
- Y位置：图层距离Y轴的位置；
- 还有一些3D视图转化和动画操作


操作
序号
f8d387c4-a8ae-4fae-9302-a
X位置
1003.86
Y位置
772.2
宽度
100
高度
40
透视
缩放
X旋转度
Y旋转度
Z旋转度
进入动画
点击查看动画类型

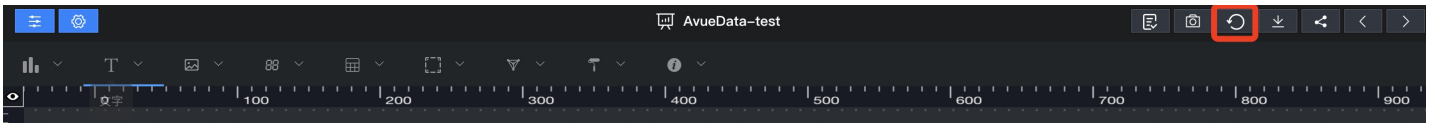
5.预览、保存组件

一、 设计好后，可以点击操作栏中的

”按钮，进行预览



二、如果想继续回到编辑页面，直接点  按钮，直接回到编辑页面



基础组件

1.背景配置

2.图表类组件

3.文本类组件

4.图片类组件

5.指标类组件

6.表格类组件

7地图类组件

8万能组件

9万能dataV组件

1.背景配置

点击屏幕，在画布右侧可看到设置屏幕样式的参数。

一、大屏名称和简介

- 大屏名称：给大屏起个名称，方便区分；
- 大屏简介：大屏的简单介绍；

图1. 11

二、大屏大小

- 大屏宽度：大屏大宽度，可根据投放显示器的屏幕大小来设置；
- 大屏高度：大屏的高度是多少，可根据投放显示器的屏幕大小来设置；

图1.12

三、大屏背景

- 背景颜色：大屏的背景颜色（只有没有背景图的时候，才显示出来）；
- 缩略图：大屏列表显示的图片；
- 背景图：大屏的背景图片；

图1.13

四、大屏比例

- 缩放：大屏的缩放比例，正常是1；

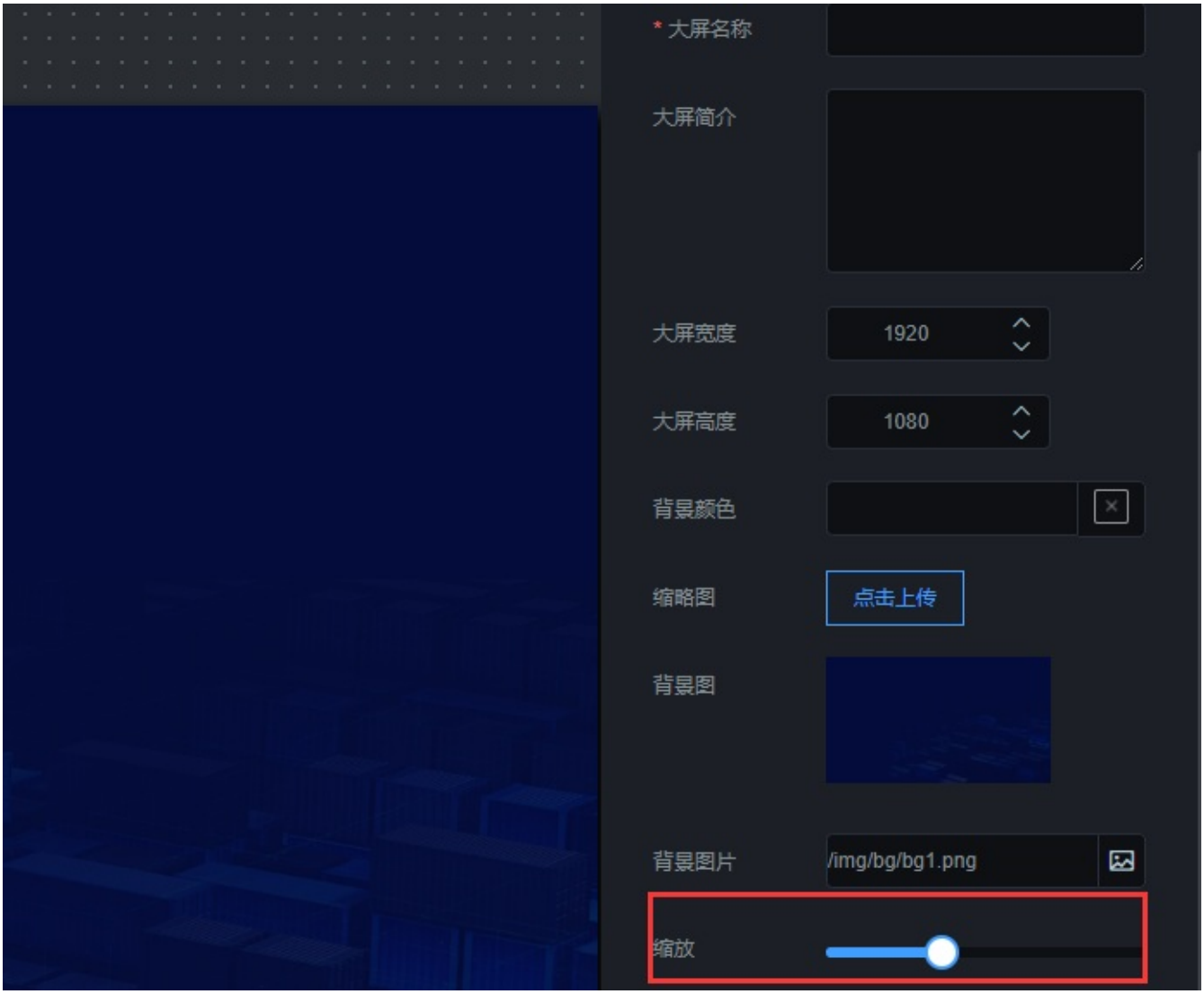


图1. 14

五、水印

如果想大屏上带上水印样式，可在这里设置，如图5.15。

- 水印开关：水印是否显示；
- 内容：水印内容；
- 大小：水印文字大小；
- 颜色：水印文字颜色；
- 角度：水印文字角度；

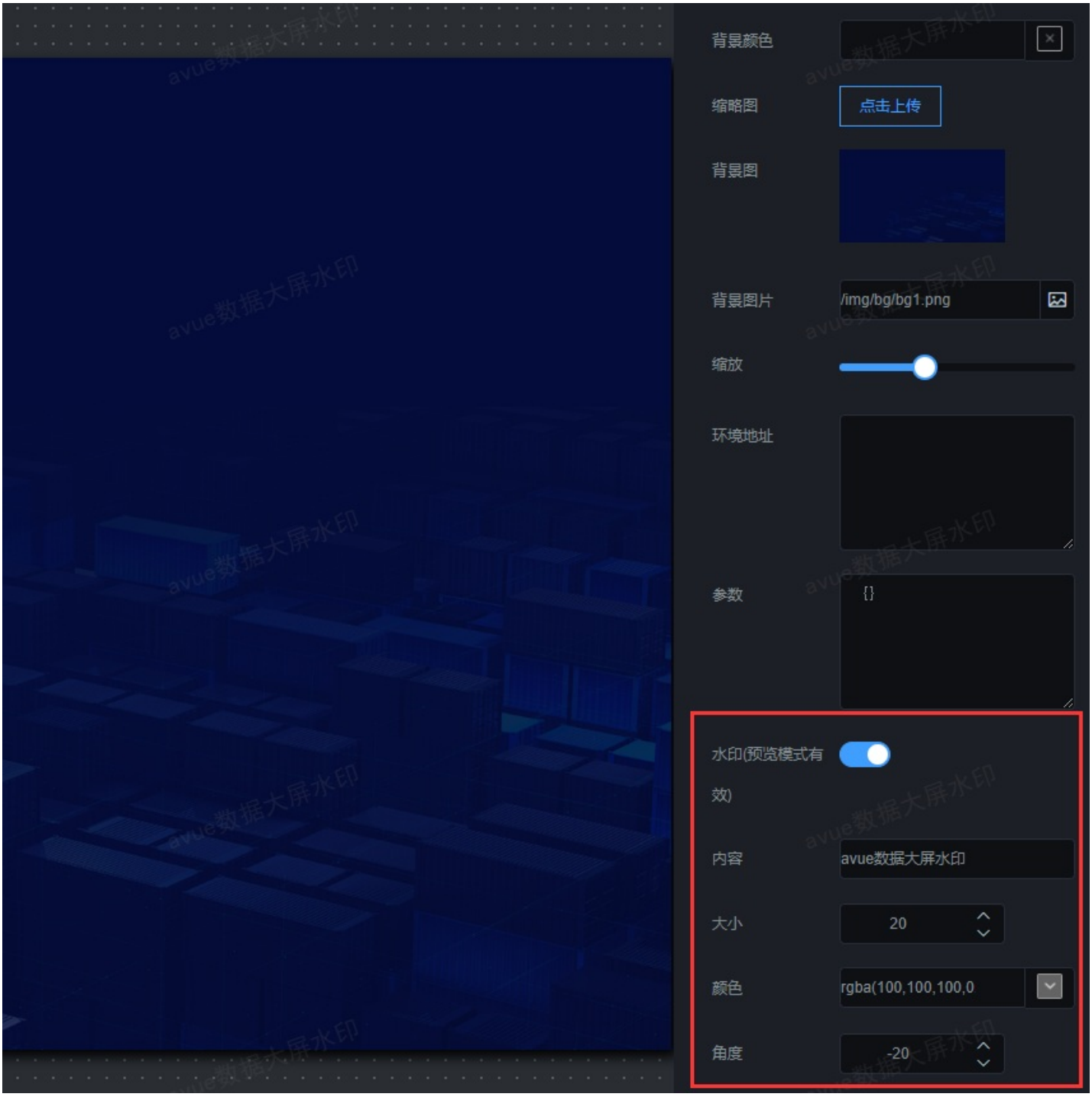

背景颜色
缩略图
点击上传
背景图
背景图片
/img/bg/bg1.png
缩放
环境地址
参数
{}
水印(预览模式有效)
内容
avue数据大屏水印
大小
20
颜色
rgba(100,100,100,0
角度
-20

图5.15

2.图表类组件

2.1柱形图组件
2.2折线图组件
2.3饼图组件
2.4环形图组件
2.5象形图组件
2.6雷达图组件
2.7散点图组件
2.8漏斗图组件

2.1柱形图组件

柱形图组件就是添加柱形图的组件。点击“  ”图标，再点击“柱形图”，即可创建新的图像，如图2.11；

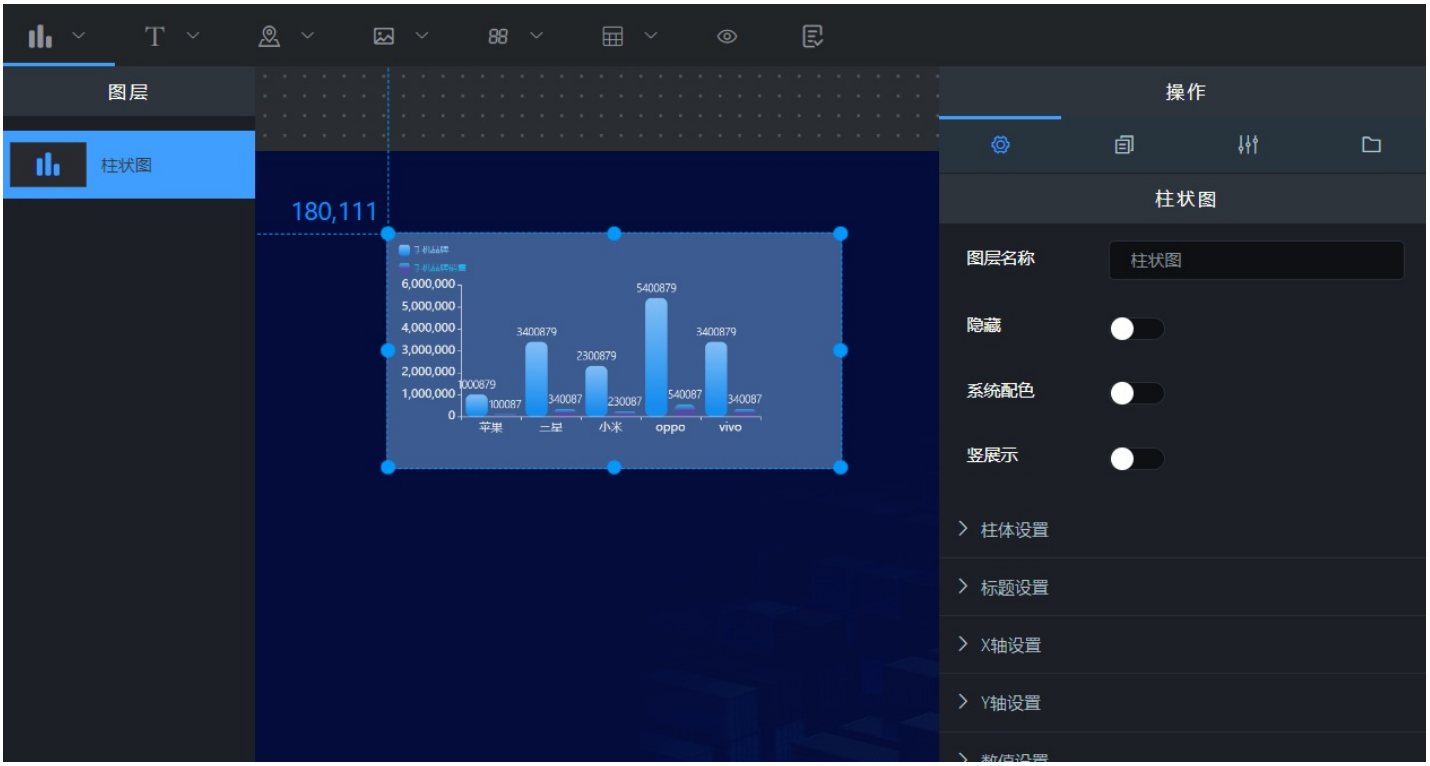


图2. 11

一、组件名称设置

选中该柱形图组件，在操作界面右侧的“图层名称”处可修改组件的名称，如图2.12。（名称最好要设置一下，方便后期组件管理）

图2. 12

二、系统配色

选中该柱形图组件，在操作界面右侧，打开“系统配色”开关，在“配色选择”下拉框中选择主题色，来修改柱形图组件的配色，如图2.13。

- 默认配色：效果图如图2.131；
- 紫色主题：效果图如图2.132；
- 绿色主题：效果图如图2.133；

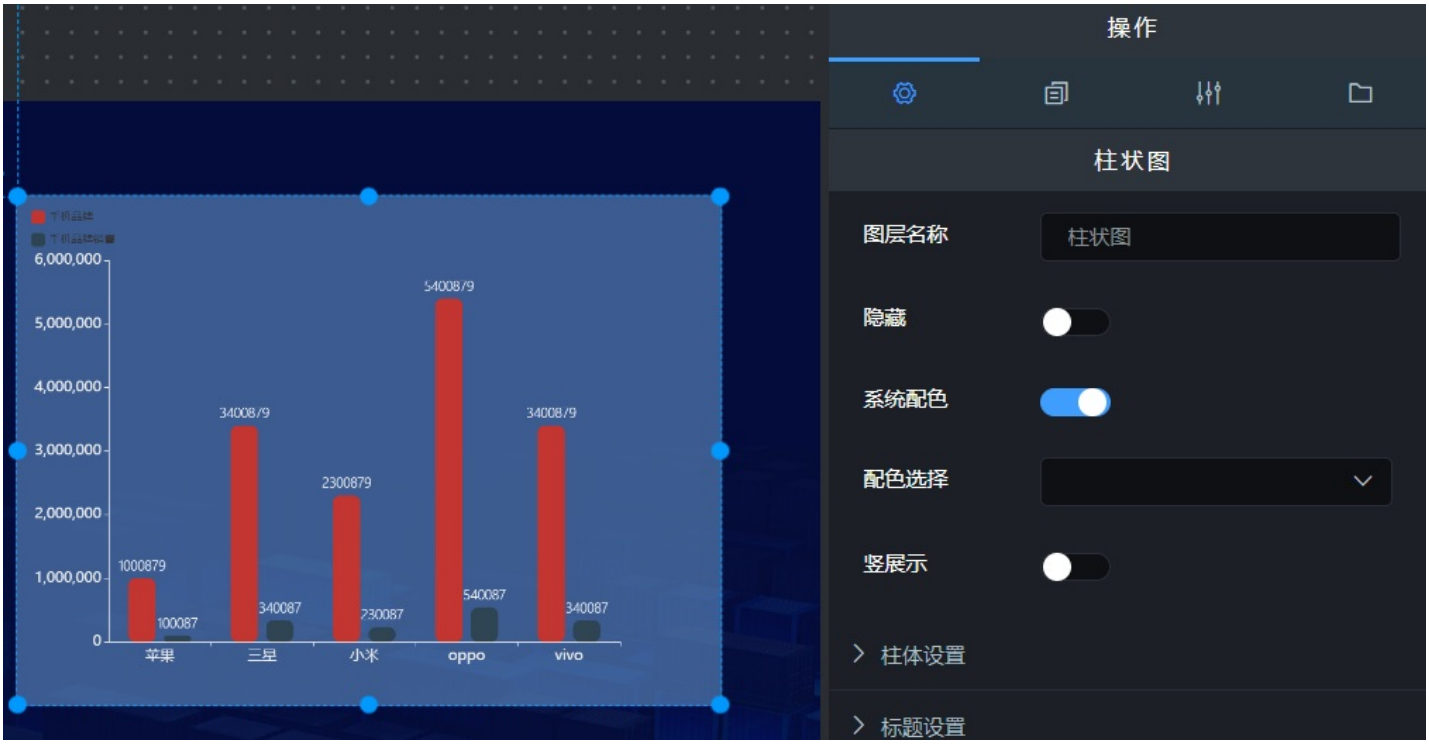


图2. 13

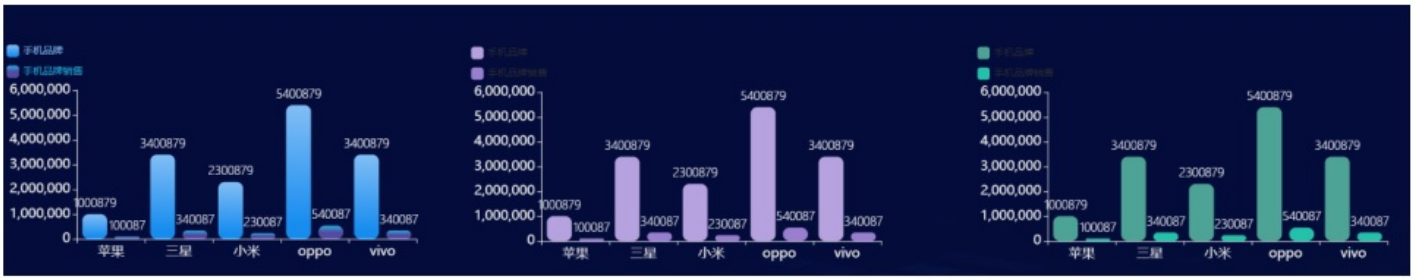


图 2.131　　图 2.132　　图 2.133

三、位置设置

想达到图2.14中图表的效果，可如下设置：

选中该柱形图组件，在操作界面右侧，打开“竖展示”开关，就可实现，操作步骤如图2.15。


3400879
vivo
340087
5400879
oppo
540087
2300879
小米
230087
3400879
三星
340087
000879
苹果
100087
0
1,000,000
2,000,000
3,000,000
4,000,000
5,000,000
6,000,000

图2. 14


操作
柱状图
图层名称
柱状图
隐藏
系统配色
竖展示
柱体设置
标题设置
X轴设置
Y轴设置
数值设置
提示语设置
坐标轴边距设置
苹果
三星
小米
oppo
vivo

图2.15

四、柱体设置

选中该柱形图组件，在操作界面右侧的“柱体设置”处可修改设置组件的外观特点，如图2.16。

- 最大宽度：柱体的最大宽度，可以调节改变柱体的宽度；
- 圆角：柱体4个角的弧度；
- 最小高度：柱体最小不能低于的高度；

图2.16

五、标题设置

选中该柱形图组件，在操作界面右侧的“标题设置”处可修改柱形图组件的标题样式，如图2.17。

- 标题开关：该开关控制标题的显示与隐藏；
- 标题：标题显示的内容；
- 字体颜色：标题的颜色；
- 字体粗细：标题字体的粗细；
- 字体大小：标题字体大小；
- 字体位置：标题的位置，分为：居中、左对齐、右对齐；
- 副标题：副标题内容（如果不想显示副标题，不填写内容就不显示）；
- 字体颜色：副标题字体颜色；
- 字体粗细：副标题字体的粗细；
- 字体大小：副标题字体大小；

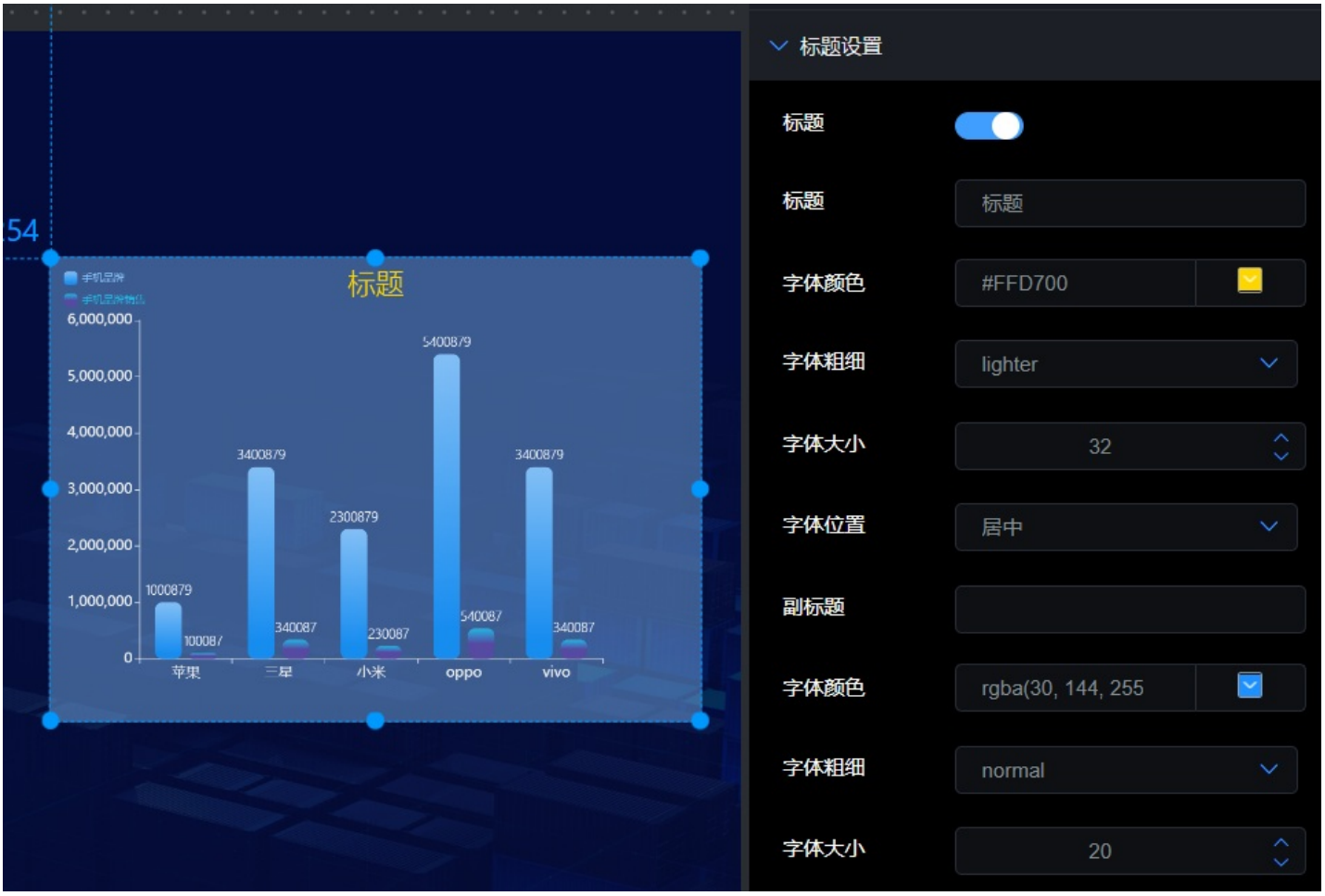


图2. 17

六、X轴设置

选中该柱形图组件，在操作界面右侧的“X轴设置”处可修改柱形图组件的X轴样式，如图2.18。

- 名称：X轴的名称；
- 显示：X轴是否显示；
- 显示网格：网格是否显示；
- 轴线颜色：网格线颜色设置；
- 标签间距：每个柱状图之间的距离；
- 文字角度：X轴文字的旋转角度；
- 轴反转：柱状图顺序左右调转；
- 字号：X轴字体大小；

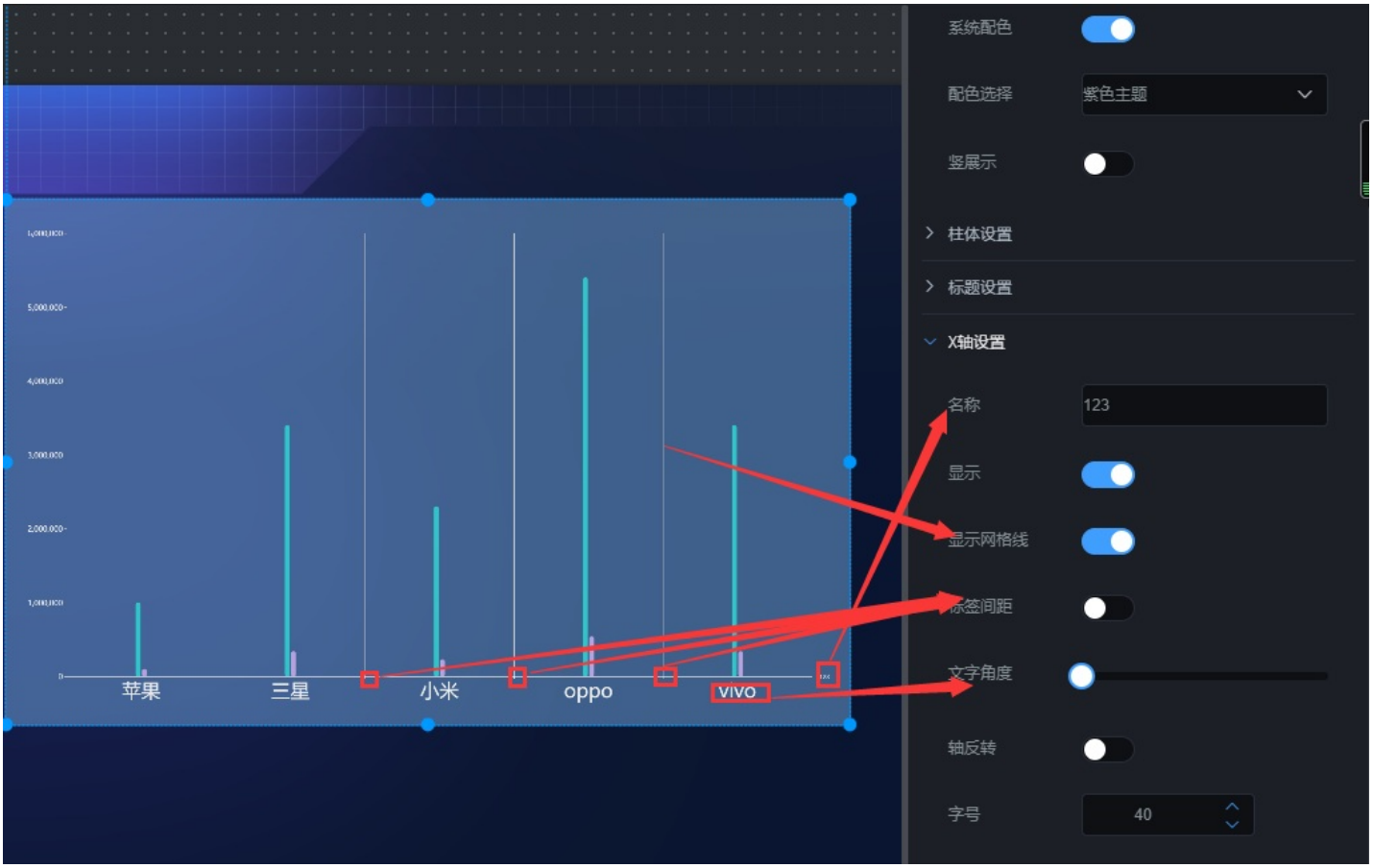


图2. 18

七、Y轴设置

选中该柱形图组件，在操作界面右侧的“Y轴设置”处可修改柱形图组件的Y轴样式，如图2.19。

- 名称：Y轴的名称；
- 显示：Y轴是否显示；
- 轴网格线：网格是否显示；
- 轴线颜色：网格线颜色设置；
- 轴反转：柱状图顺序上下调转；打开轴翻转开关，效果图如2.191；
- 字号：Y轴字体大小；

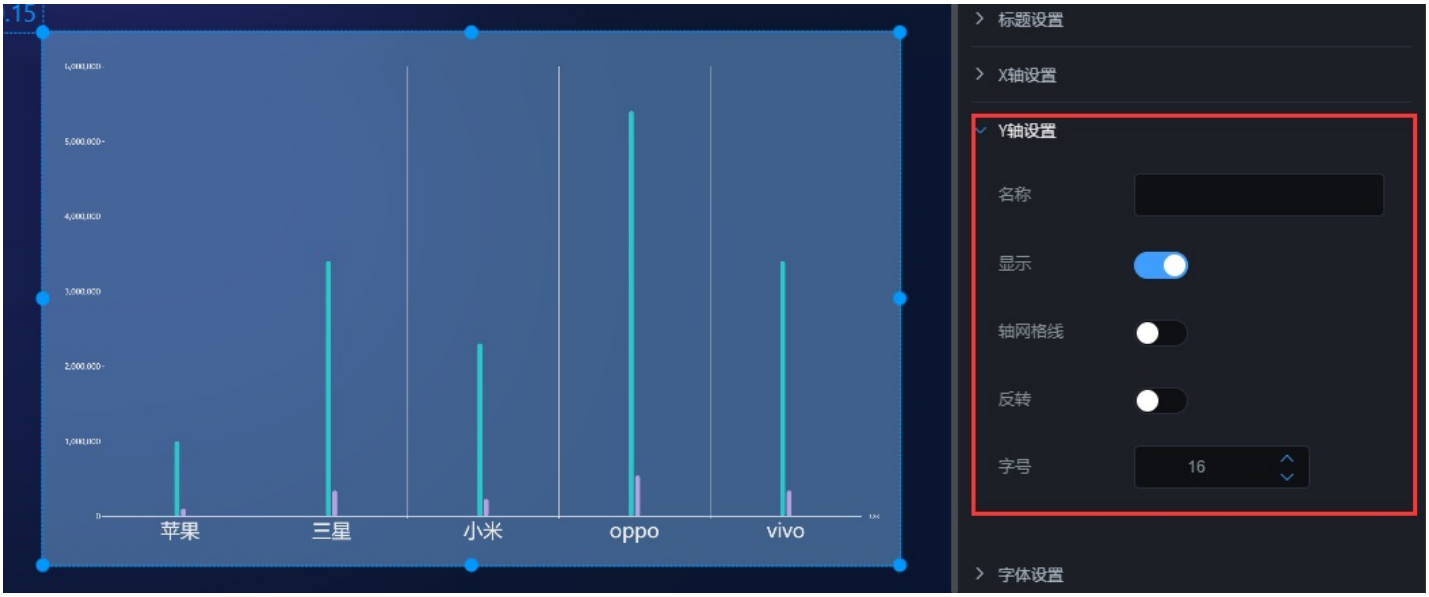


图2. 19

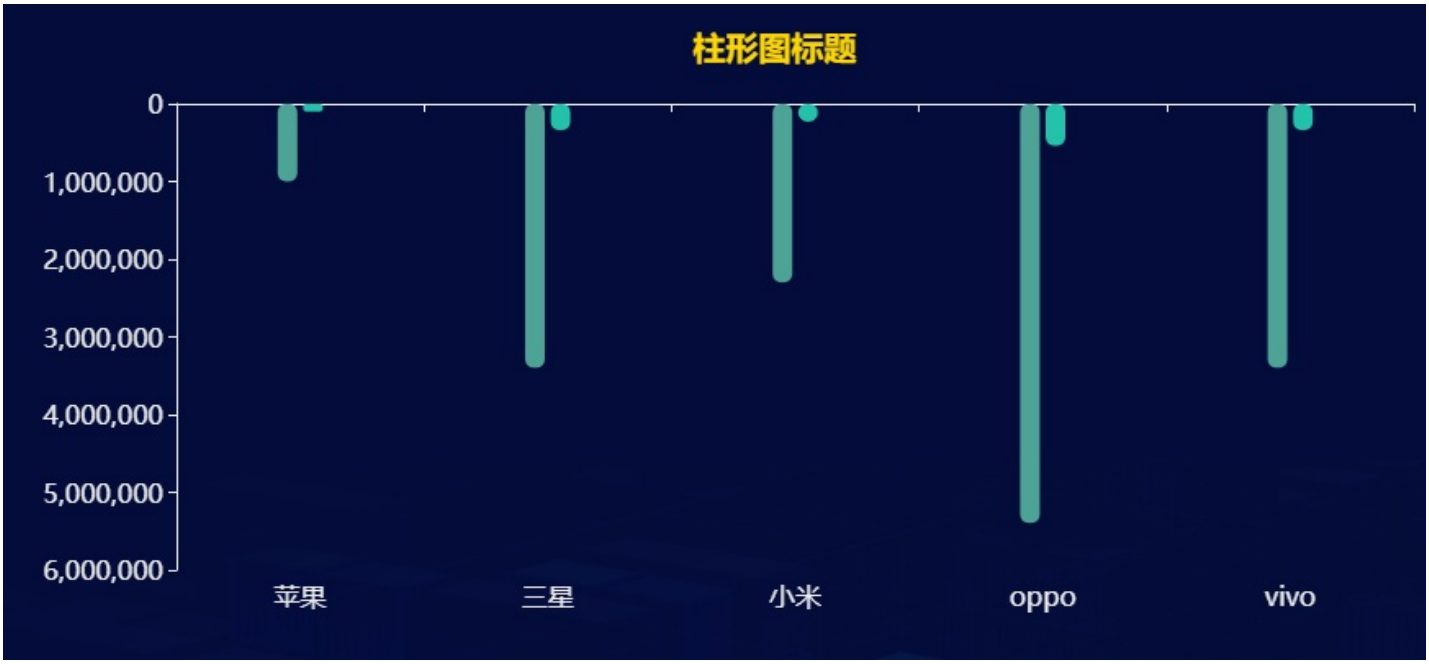


图2. 191

八、字体设置

选中该柱形图，在操作界面右侧的“字体设置”处可修改柱体顶部文字的样式，如图2.192。

- 显示：数值文字是否显示；
- 字体大小：文字的大小；
- 字体颜色：文字的颜色；
- 字体粗细：文字的粗细；

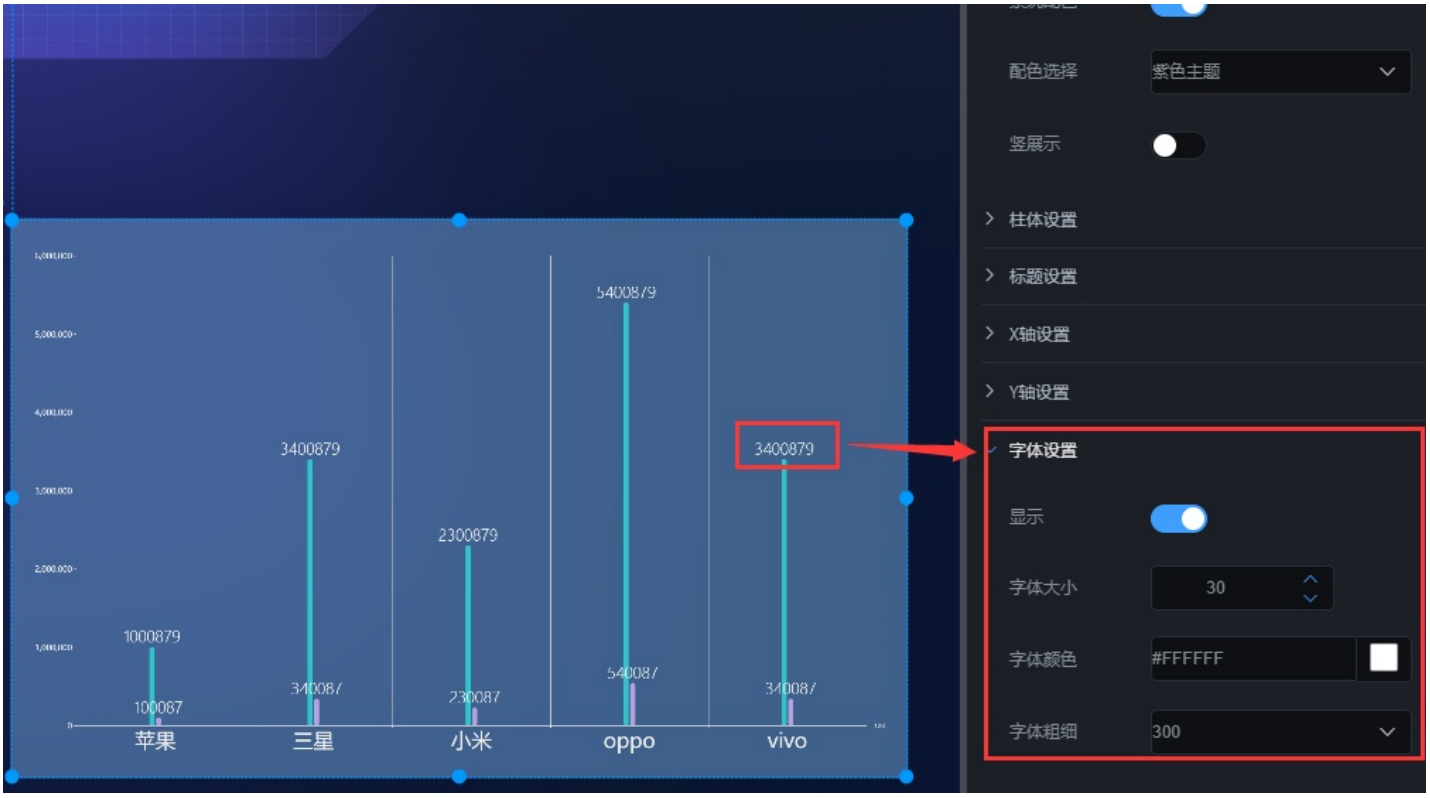


图2. 192

九、提示语设置

选中该柱形图组件，在操作界面右侧的“提示语设置”处可修改柱形图组件的提示语，如图2.193。效果图如图2.194；

- 字体大小：提示语的字体大小；
- 字体颜色：提示语的字体颜色；

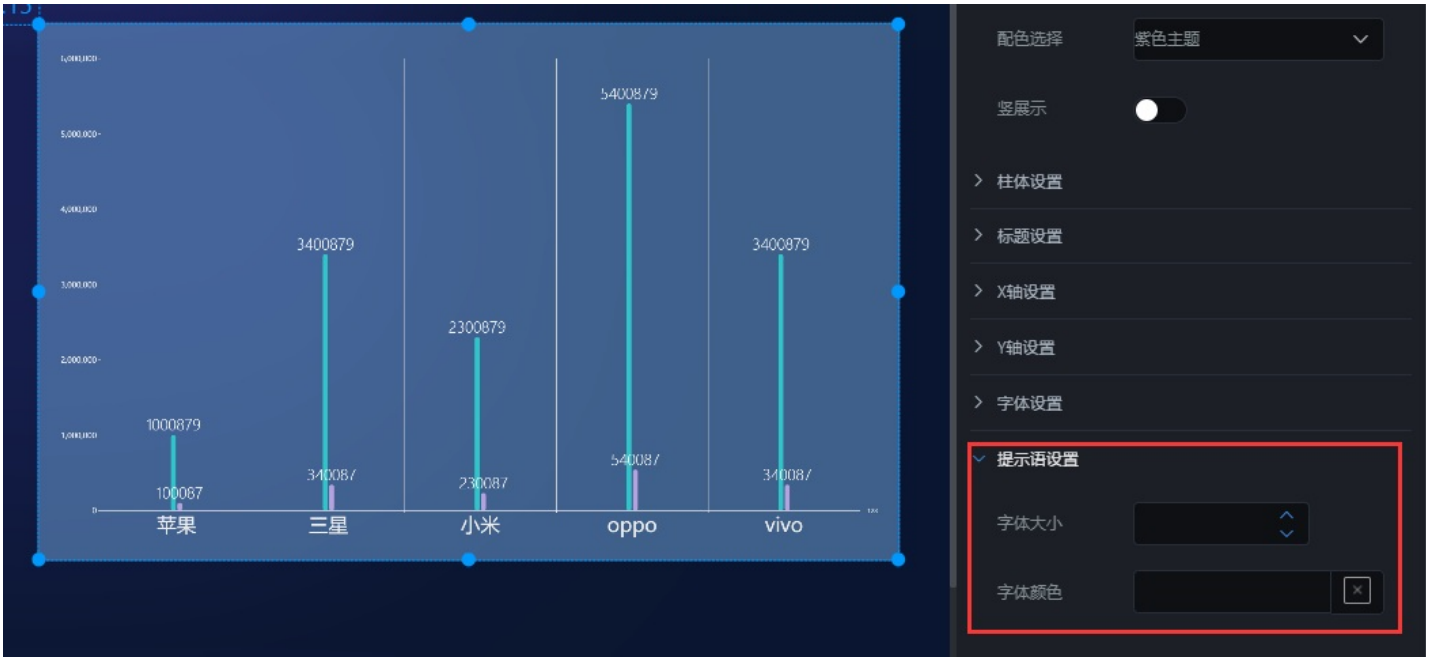


图2. 193

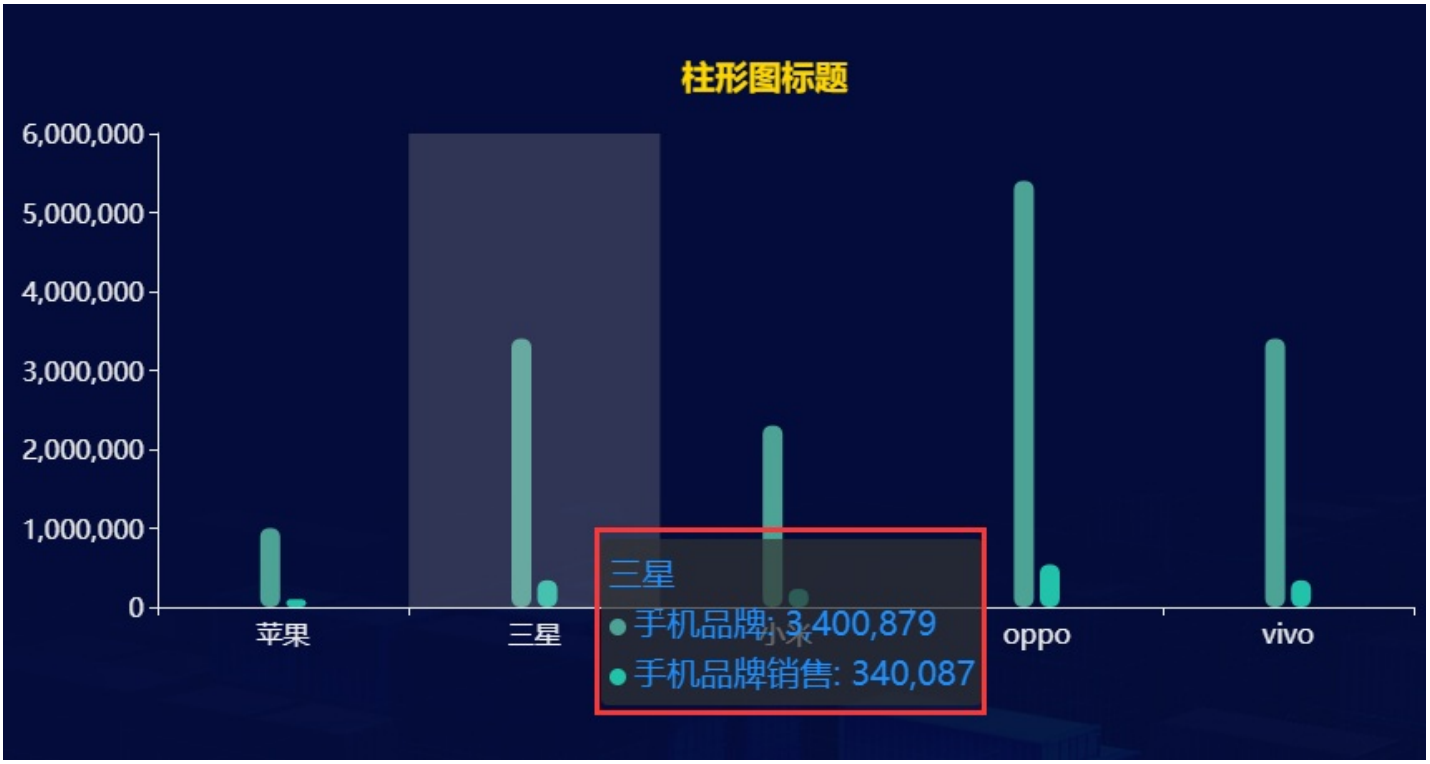


图2. 194

十、坐标轴边距设置

选中该柱形图组件，在操作界面右侧的“坐标轴边距设置”处可修改柱形图距离左、右、上、下的距离，如图2.195。

- 左边距（像素）：柱形图距离左边的距离；
- 顶边距（像素）：柱形图距离顶部的距离；
- 右边距（像素）：柱形图距离右边的距离；
- 底边距（像素）：柱形图距离底部的距离；

图2. 195

十一、图例设置

选中该柱形图组件，在操作界面右侧的“图例设置”处可设置图例的样式，如图2.196；效果图如2.197。

- 图例开关：是否显示图例；
- 字体颜色：图例的字体颜色。如果要想自定义图例的颜色，需要关闭系统配色（图2.1971）和删除所有自定义配色（图2.1972）中的颜色。
- 图例宽度：图例的宽度；
- 横向位置：图例的位置，分为：居中、左对齐、右对齐，如图2.1973；
- 纵向位置：图例的位置，分为：顶部、底部，如图2.1974；
- 布局朝向：图例的排列顺序，分为：横排和竖排，如图2.1975；
- 字体大小：图例的字体大小；

图2. 196

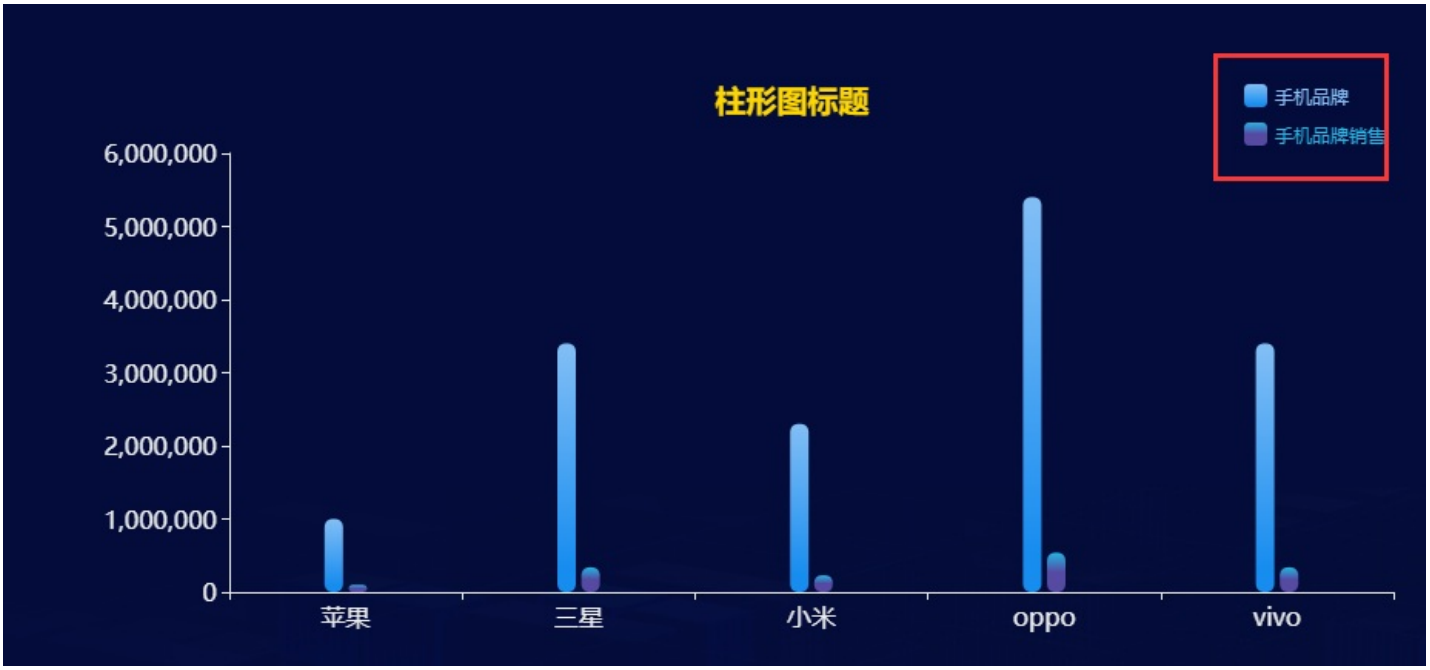
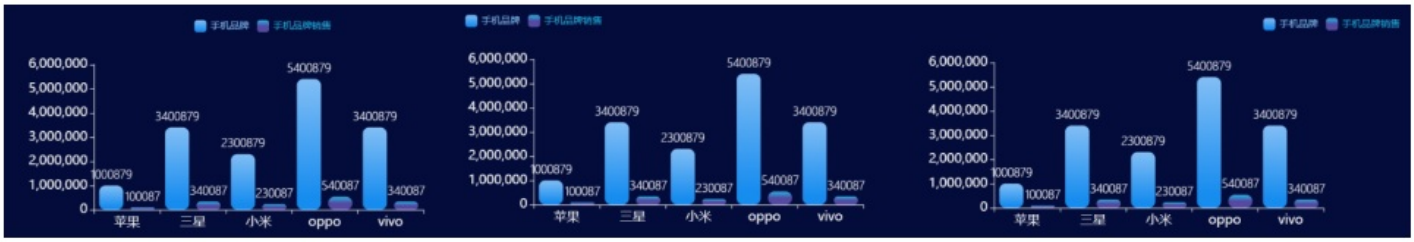


图2.197

图2.1971

图2. 1972



居中　　左对齐　　右对齐

图2. 1973

顶部　　底部

图2. 1974

竖排　　横排

图2. 1975

十二、自定义配色设置

选中该柱形图组件，在操作界面右侧的“自定义配色设置”处可配置上边不能设置的内容，如图2.198。

- 文字颜色：X、Y轴字体的颜色；
- 轴线颜色：X、Y轴轴线颜色；
- 配色：柱体的颜色；如果开启了“系统配色”，需要先把系统配色先关掉，这样自定义的柱体颜色才起作用；

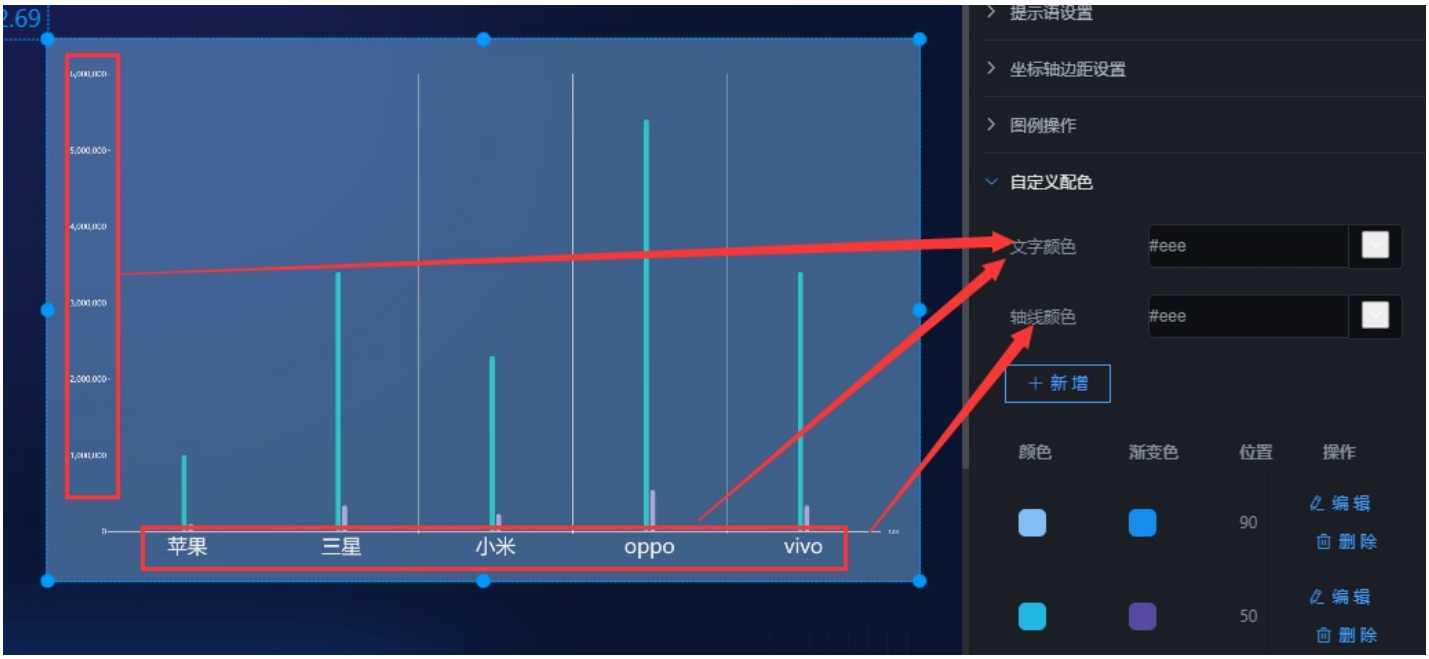


图2. 198

十三、接口设置

选中该柱形图组件，在操作界面右侧，点击“

”，可设置接口，如图2.199。

1. 数据类型

数据类型分为静态数据和动态数据；

- 静态数据：写死的数据；
- 动态数据：会随着接口传过来的数据实时变化；（一般这种比较常用）

2. 接口地址

- categories：X轴参数；
- series：柱形内容；内部 name为柱形名称，data为柱形相对应Y轴的值；

（1）静态数据，接口地址传过来的内容要符合以下格式：

```
（1）两个柱形图接口格式：
{"categories":["苹果","三星","小米","oppo","vivo"],"series":[{"name":"手机品牌","d
ata":[1000879,3400879,2300879,5400879,3400879]},{"name":"手机品牌销售","data":[10
```

```
0087,340087,230087,540087,340087]}]}

（2）一个柱形图接口格式：
{"categories":["苹果","三星","小米","oppo","vivo"],"series":[{"name":"手机品牌","d
ata":[1000879,3400879,2300879,5400879,3400879]}]}
```

（2）动态数据，接口地址传过来的内容要符合以下格式：

```
（1）两个柱形图接口格式如下，效果图如图2.1991；
{"data":{"categories":["苹果","三星","小米","oppo","vivo"],"series":[{"name":"手
机品牌","data":[1000879,3400879,2300879,5400879,3400879]},{"name":"手机品牌销售","
data":[100087,340087,230087,540087,340087]}]}}

（2）一个柱形图接口格式如下，效果图如图2.1992；
{"data":{"categories":["苹果","三星","小米","oppo","vivo"],"series":[{"name":"手
机品牌","data":[1000879,3400879,2300879,5400879,3400879]}]}}
```

3. 刷新时间

这个参数主要针对动态数据设置的，完成数据的实时更新。

- 如果你想设置成5秒刷新一次，可以将刷新时间设置成“5000”；

4. 刷新数据

这个参数主要是重新请求以下接口，完成数据的更新。


数据类型
静态数据
动态数据
接口地址
接口方式
POST
GET
接口参数
编辑
刷新时间
5000
数据处理
编辑
刷新

图2. 199

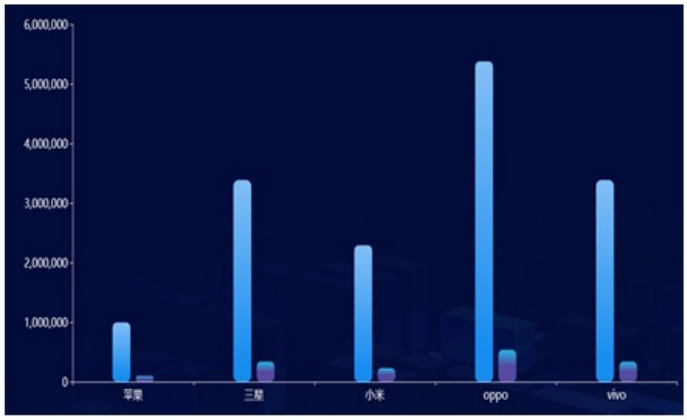


图 2.1991

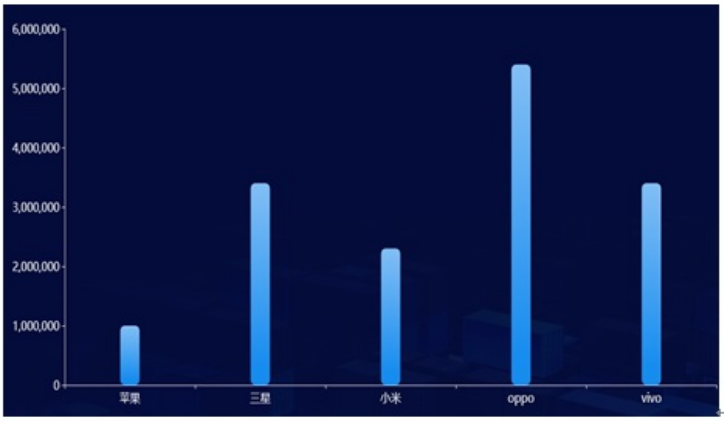


图 2.1992

5. 接口参数

暂时未用到；

6. 数据处理

这个参数用于处理图表中的数据，比如：你在不方便修改接口的时候，想修改一下图表中的某一个或某几个数据，可在这里边修改。

- 查看数据格式：

点击数据“数据处理”的编辑按钮，输入“console.log('======>',option)”，如图2.1993操作；再点击“确定”按钮，再点击“刷新”，打开调试模式，就可看到数据格式和数

据内容，可根据自己需求修改，如图2.1994；

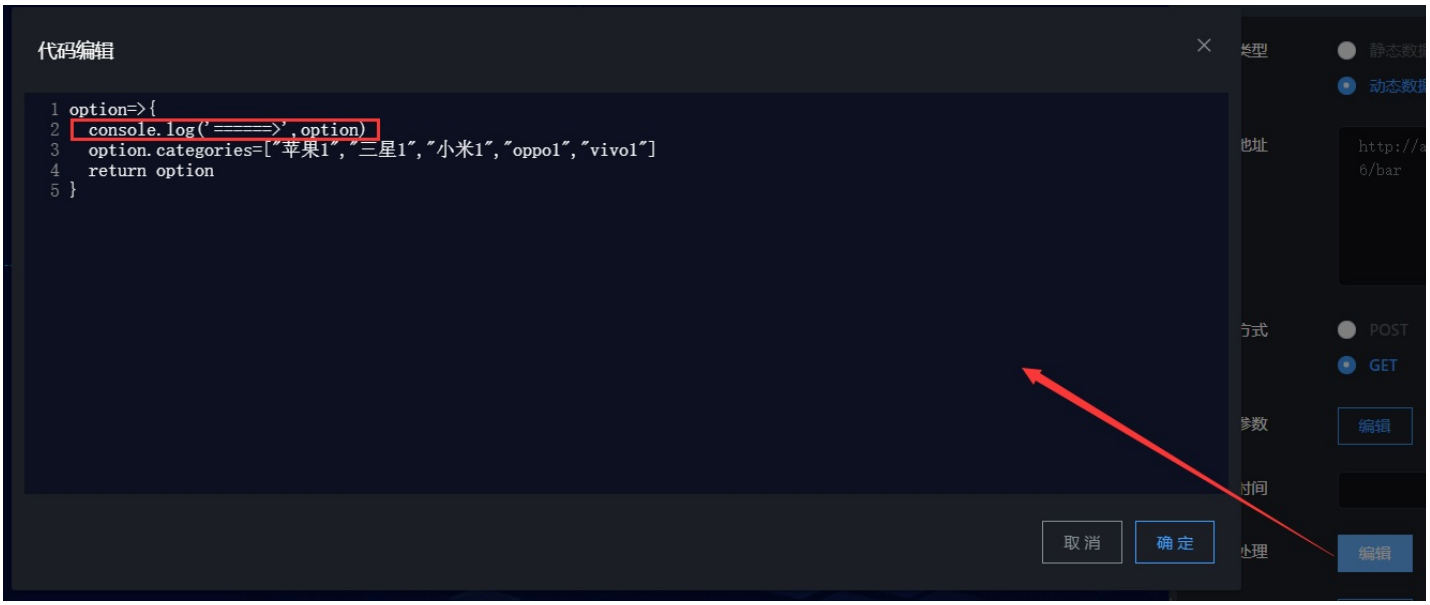


图2.1993

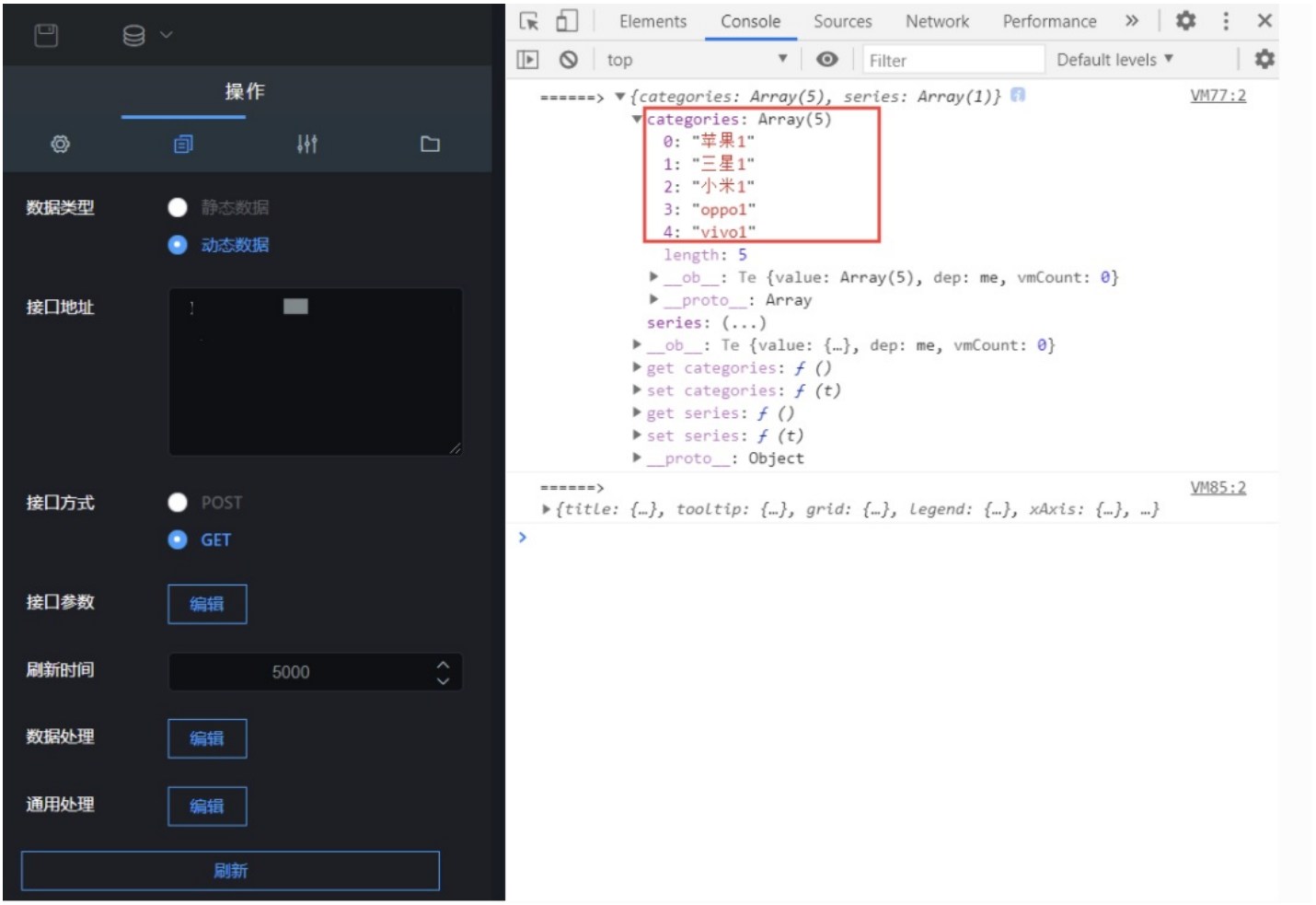


图2.1994

- 修改格式如下：

```
option=>{
  option.categories=["苹果1","三星1","小米1","oppo1","vivo1"]
  return option
}
```

7.通用处理

这个参数用于处理图表中的样式，比如：你在不方便修改的样式或样式配置中没有，可在这里边修改。

- 查看数据格式：

点击数据“通用处理”的编辑按钮，输入“console.log('=======>',option)” ，如图2.1995操作；再点击“确定”按钮，再点击“刷新” ，打开调试模式，就可看到数据格式和数据内容，可根据自己需求修改，如图2.1996；

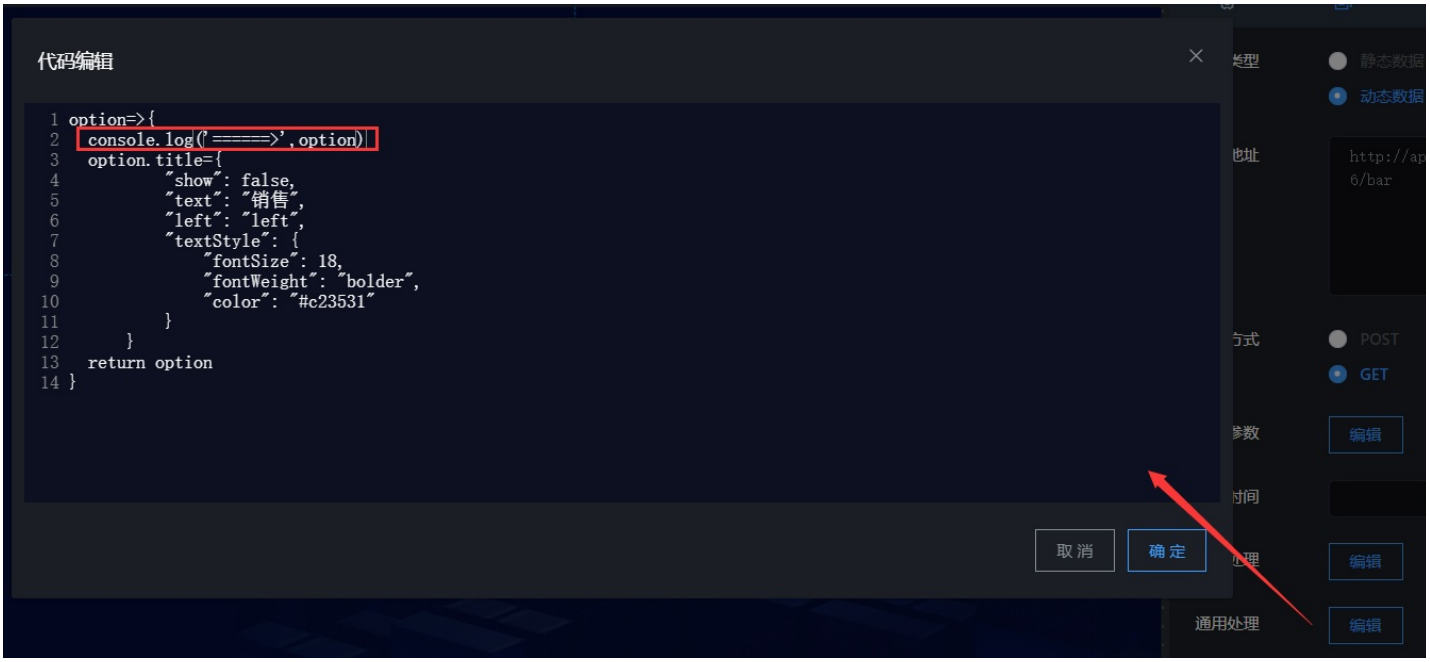


图2.1995

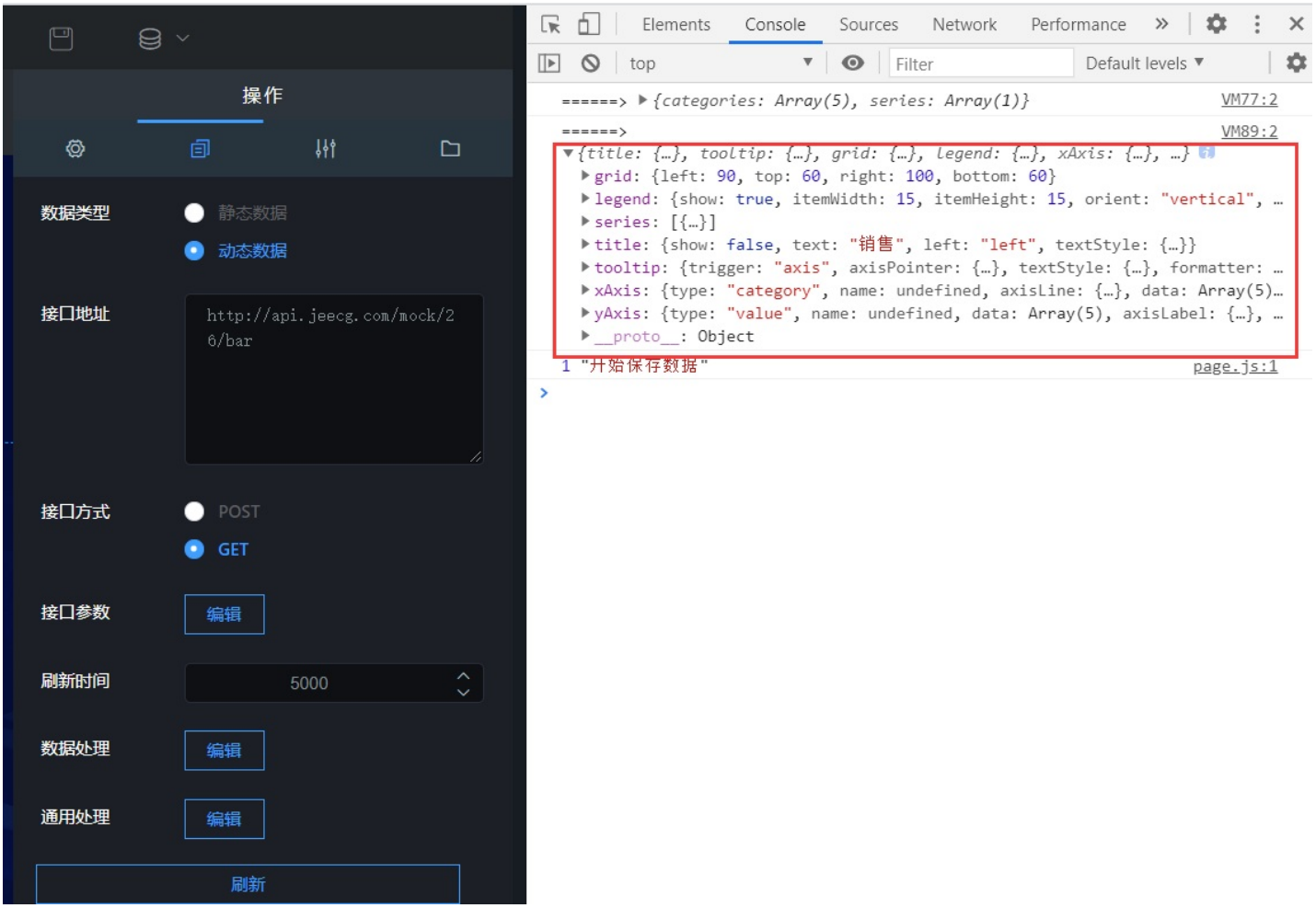


图2.1996

- 修改格式如下：

```
option=>{
```

```
  option.title={
          "show": false,
          "text": "销售",
          "left": "left",
          "textStyle": {
              "fontSize": 18,
              "fontWeight": "bolder",
              "color": "#c23531"
          }
      }
  return option
}
```

2.2折线图组件

折线图组件就是添加折线图的组件。点击“ ”图标，再点击“折线图”，即可创建新的图像，如图2.21；

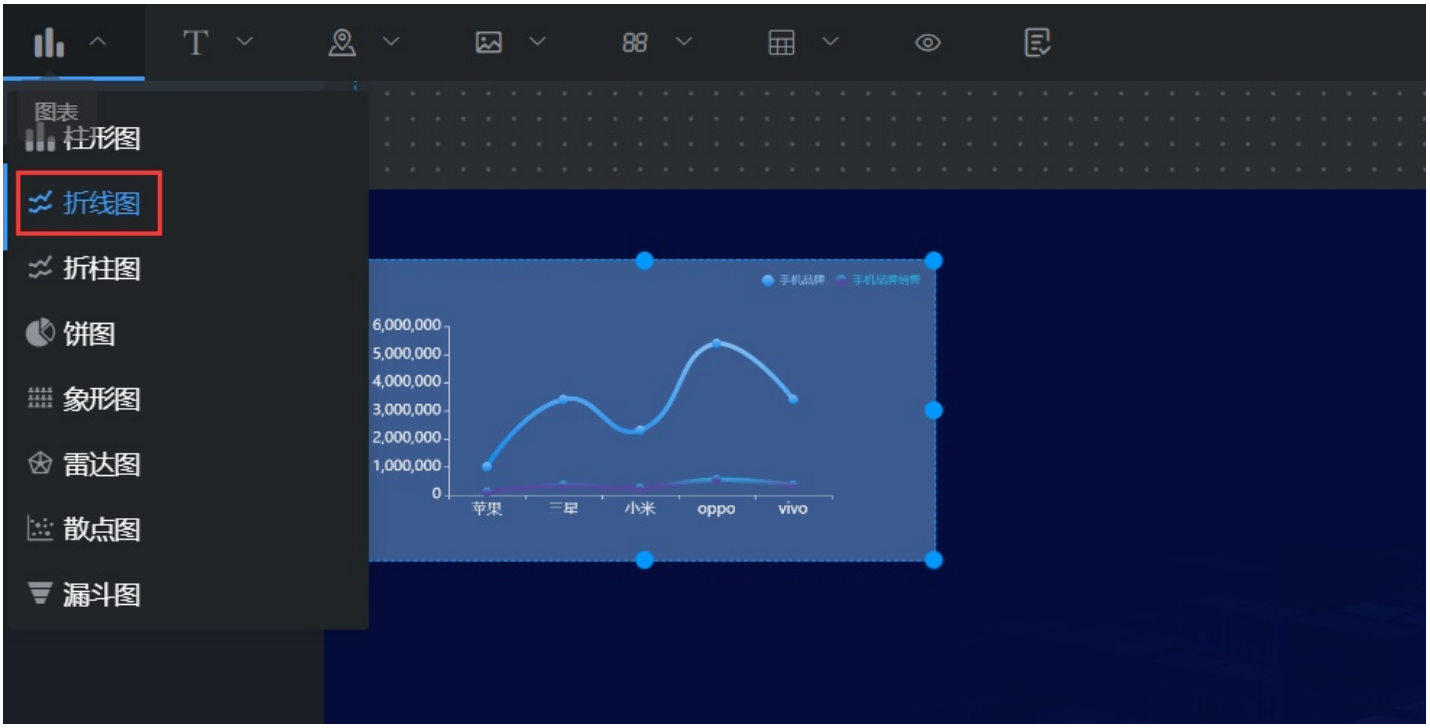


图2.21

一、组件名称设置

选中折线图组件，在操作界面右侧的“图层名称”处可修改组件的名称，如图2.22。（名称最好要设置一下，方便后期组件管理）

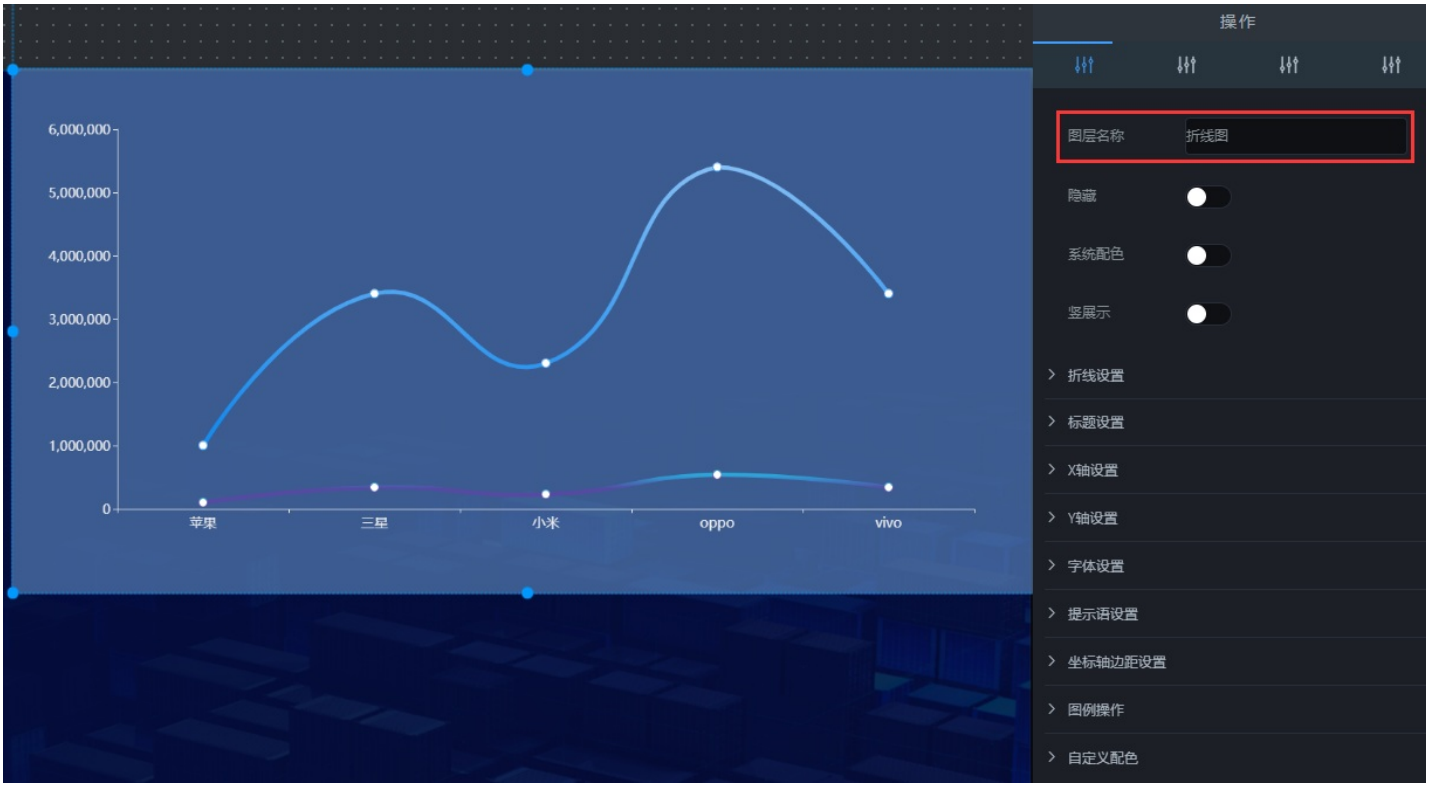


图2. 22

二、系统配色

选中该折线图组件，在操作界面右侧，打开“系统配色”开关，在“配色选择”下拉框中选择主题，来修改折线图组件的配色，如图2.23。

- 默认配色：效果图如图2.231；
- 紫色主题：效果图如图2.232；
- 绿色主题：效果图如图2.233；

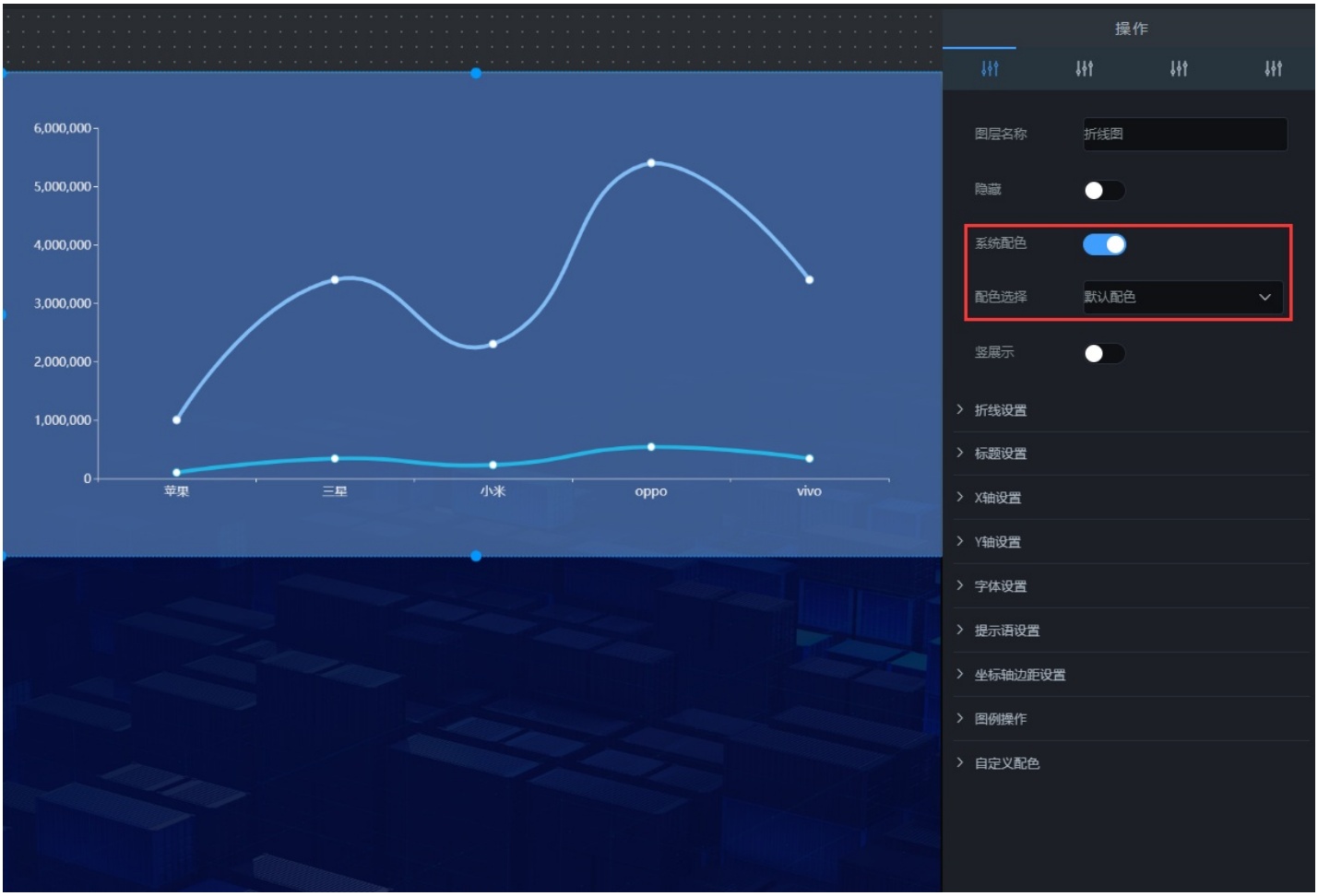


图2. 23

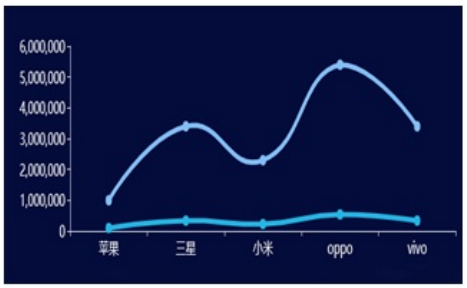


图 2.231

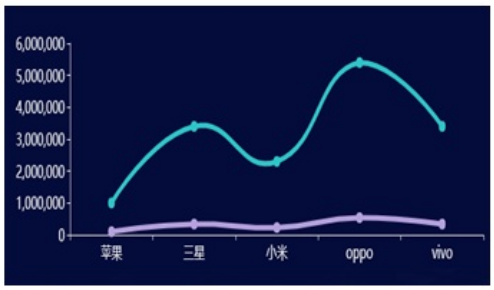


图 2.232

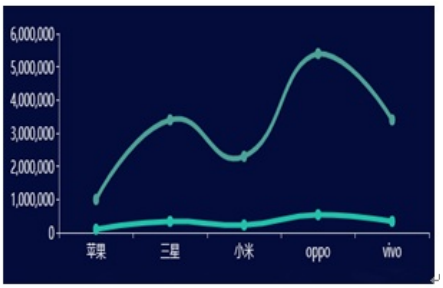


图 2.233

三、位置设置

选中该折线图组件，在操作界面右侧，打开“竖展示”开关，可修改折线图组件的显示方向，如图2.24；效果图如图2.241。

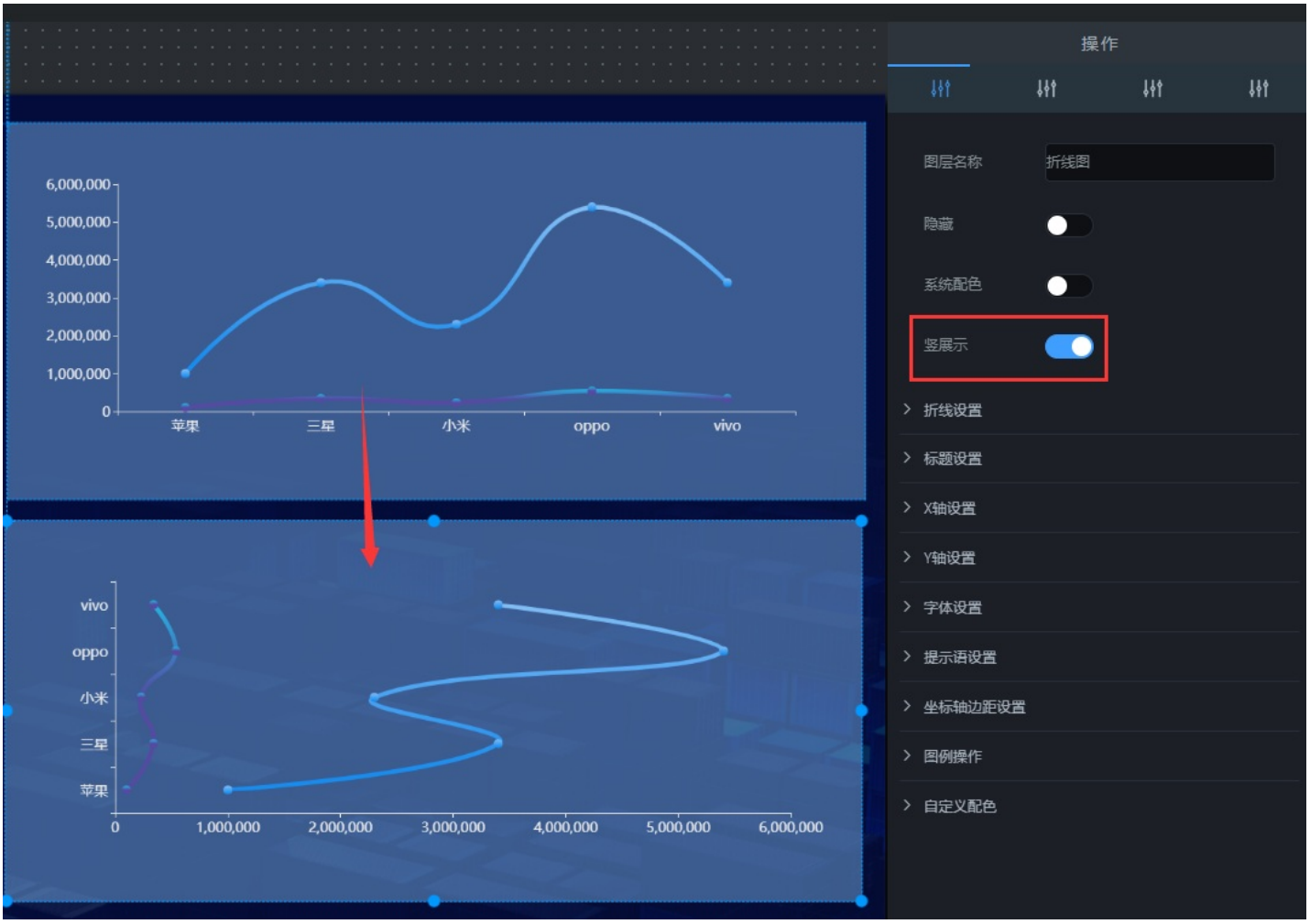


图2. 24

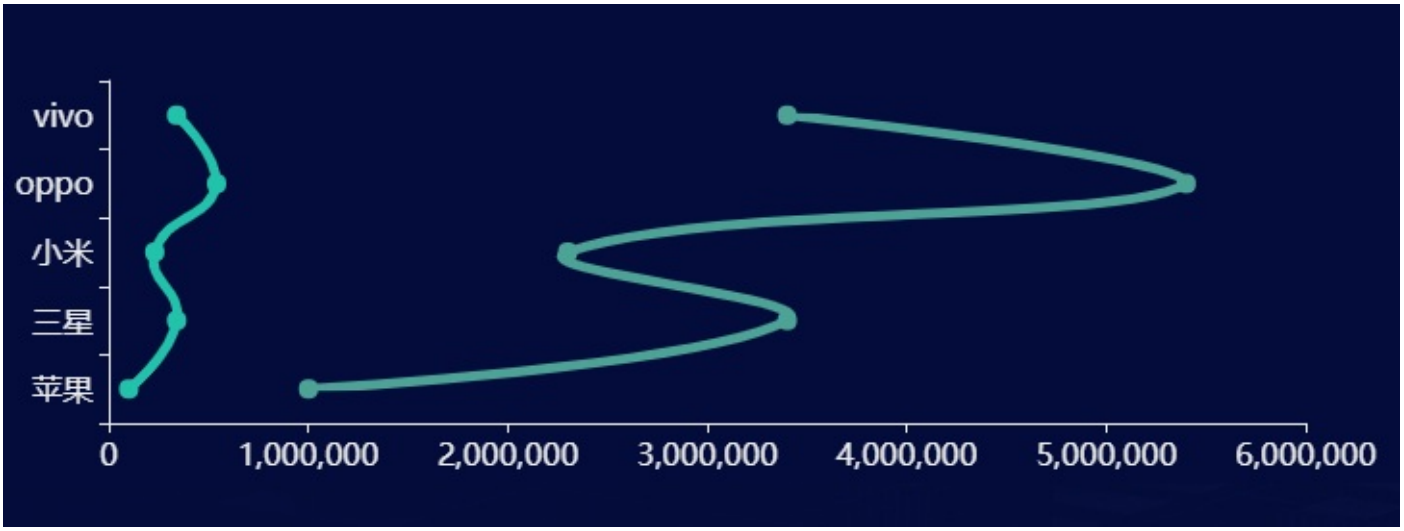


图2. 241

四、折线设置

选中该折线图组件，在操作界面右侧的“折线设置”处可修改设置组件的外观特点，如图2.25。

- 标记点：设置折线图上是否有标记点，默认有标记点；（关闭标记点开关，效果图如图

2.251；打开标记点开关，效果图如图2.252）

- 点大小：设置标记点的大小；
- 平滑曲面开关：设置折线图是平滑曲面还是折线，默认是平滑曲面；（关闭平滑曲面开关，效果图如图2.253；打开平滑曲面开关，如图2.254）
- 阶梯线图：包含当前点、中间点、下个拐点。当前点指：当前点做阶梯拐点，如图2.255；中间点指：当前为中间点做拐点，如图2.256；下个拐点：下个点做拐点，如图2.257。
- 面积堆积：设置折线图是否是面积堆积样式；（关闭面积堆积开关，效果图图如图2.258，打开面积堆积开关，效果图如图2.259）
- 线条宽度：设置线条的宽度；（图2.25标记1）

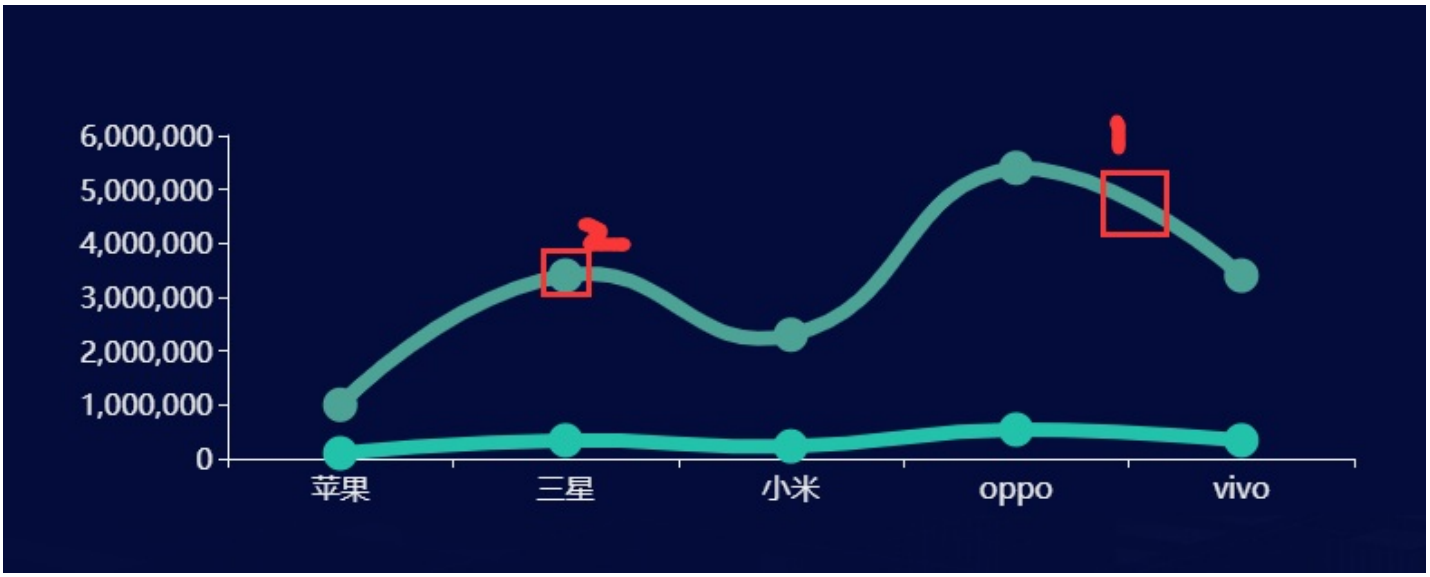


图2. 25

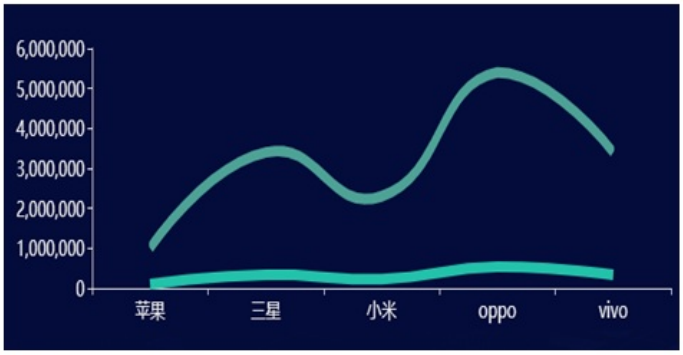


图 2.251

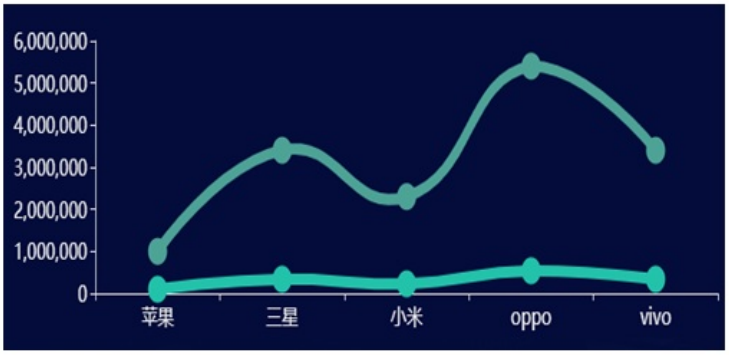


图 2.252

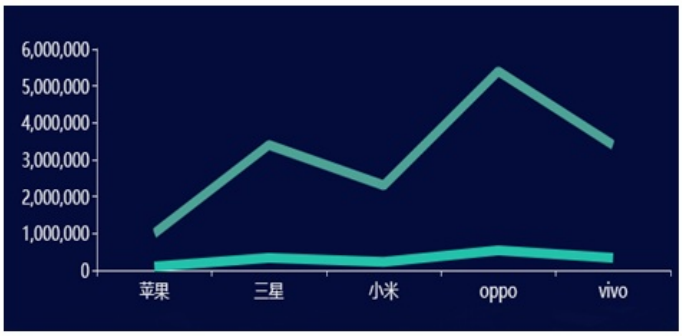


图 2.253

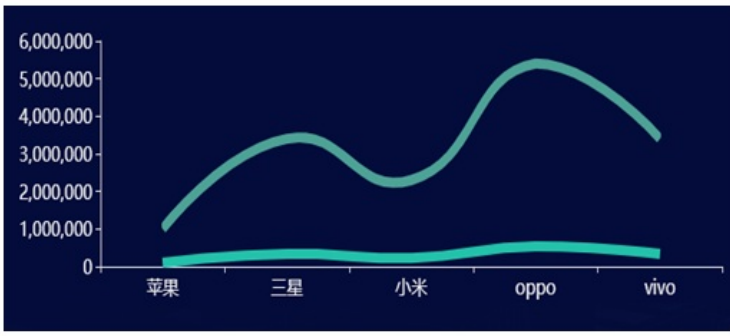


图 2.254

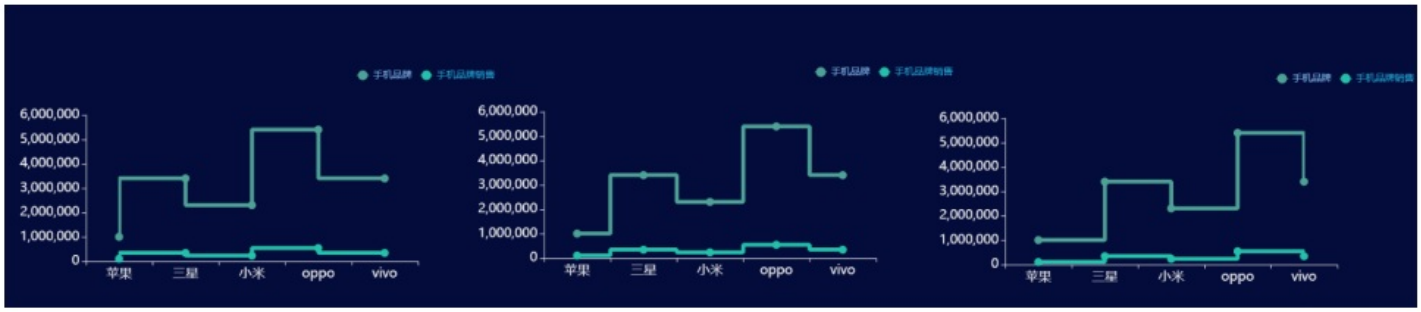


图 2.255

图 2.256

图 2.257

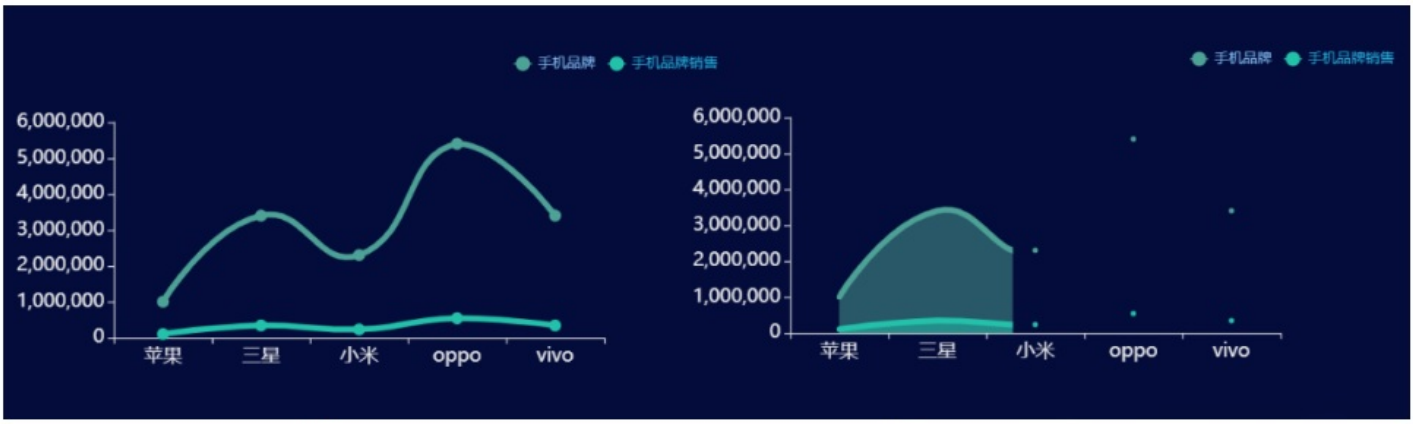


图 2.258

图 2.259

五、标题设置

选中该折线图组件，在操作界面右侧的“标题设置”处可修改折线图组件的标题样式，如图2.26。

- 标题开关：该开关控制标题的显示与隐藏；
- 标题：标题显示的内容；
- 字体颜色：标题的颜色；
- 字体粗细：标题字体的粗细；
- 字体大小：标题字体大小；
- 字体位置：标题的位置，分为：居中、左对齐、右对齐；
- 副标题：副标题内容；

- 字体颜色：副标题字体颜色；
- 字体粗细：副标题字体的粗细；
- 字体大小：副标题字体大小；

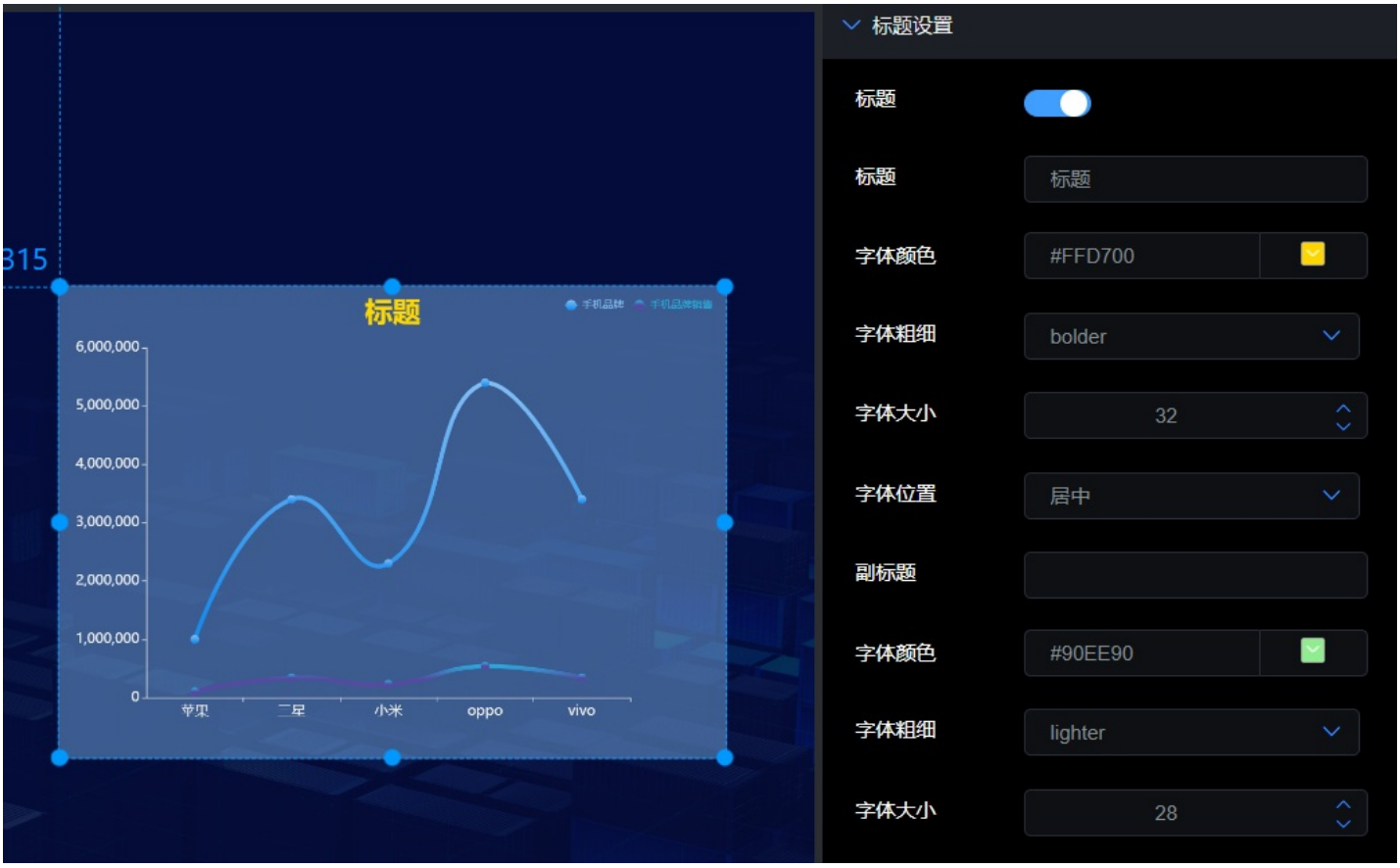


图2. 26

六、X轴设置

选中该折线图组件，在操作界面右侧的“X轴设置”处可修改折线图组件的X轴样式，如图2.27。

- 名称：X轴的名称；
- 显示：X轴是否显示；
- 显示网格：网格是否显示；（打开显示网格线开关，效果图如图2.271）
- 轴线颜色：网格线颜色设置；
- 标签间距：每个类型之间的距离；
- 文字角度：X轴文字的旋转角度；（调整文字角度，效果图如图2.272）
- 轴反转：折线图顺序左右调转；
- 字号：X轴字体大小；

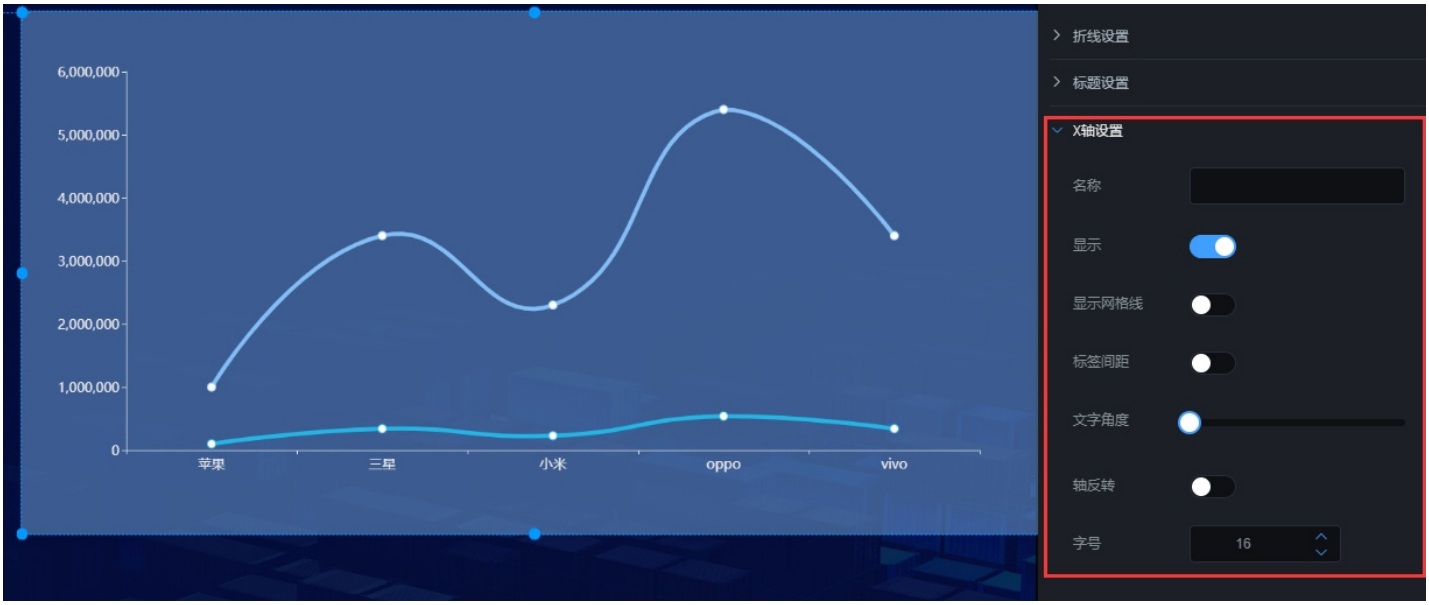


图2. 27

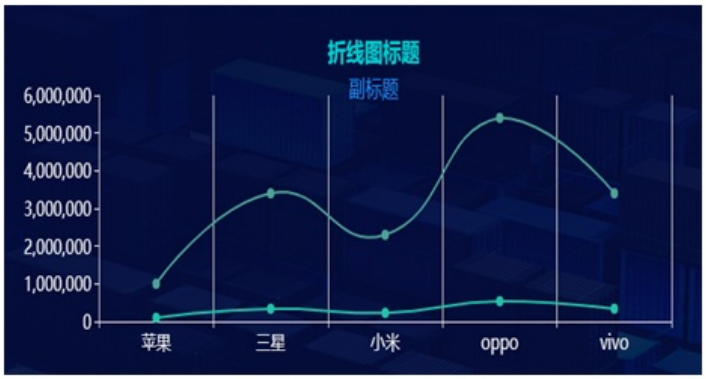


图 2.271

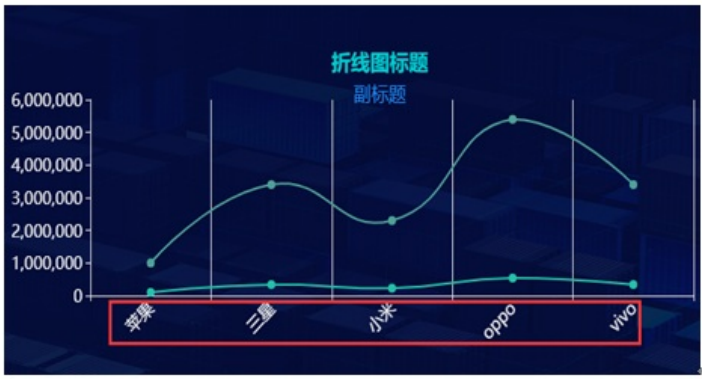


图 2.272

七、Y轴设置

选中该折线图组件，在操作界面右侧的“Y轴设置”处可修改折线图组件的Y轴样式，如图2.28。

- 名称：Y轴的名称；
- 显示：Y轴是否显示；
- 轴网格线：网格是否显示；
- 轴线颜色：网格线颜色设置；
- 轴反转：折线图顺序上下调转；
- 字号：Y轴字体大小；

图2. 28

八、字体设置

选中该折线图，在操作界面右侧的“字体设置”处可修改折线图组件上文字的样式，如图2.29。

备注：上边“折线设置 ->标记点”开关一定打开

- 显示：数值文字是否显示；
- 字体大小：文字的大小；
- 字体颜色：文字的颜色；
- 字体粗细：文字的粗细；

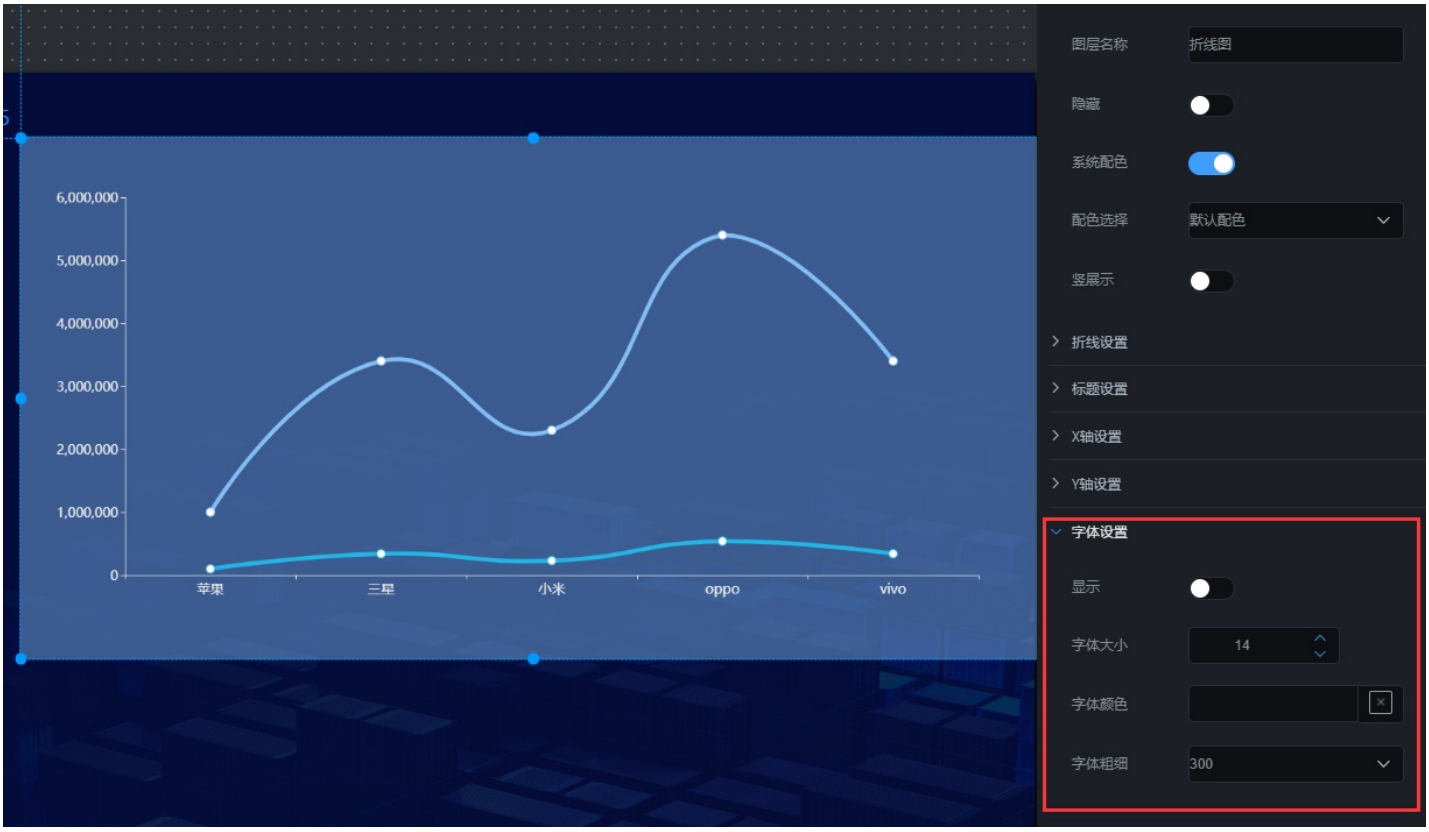


图2. 29

九、提示语设置

选中该折线图组件，在操作界面右侧的“提示语设置”处可修改折线图组件的提示语，如图2.291；效果图如图2.292；

- 字体大小：提示语的字体大小；
- 字体颜色：提示语的字体颜色；

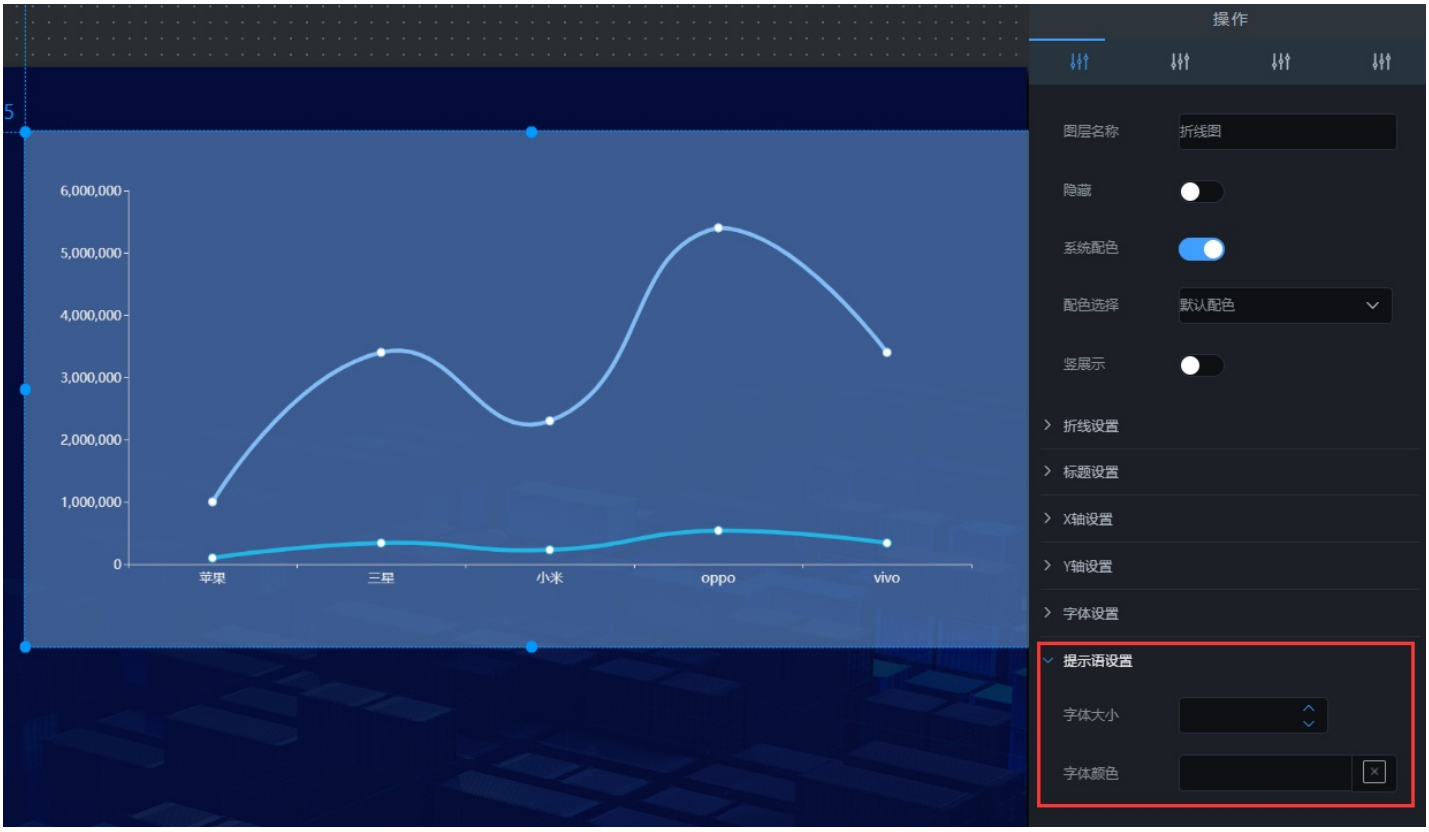


图2. 291

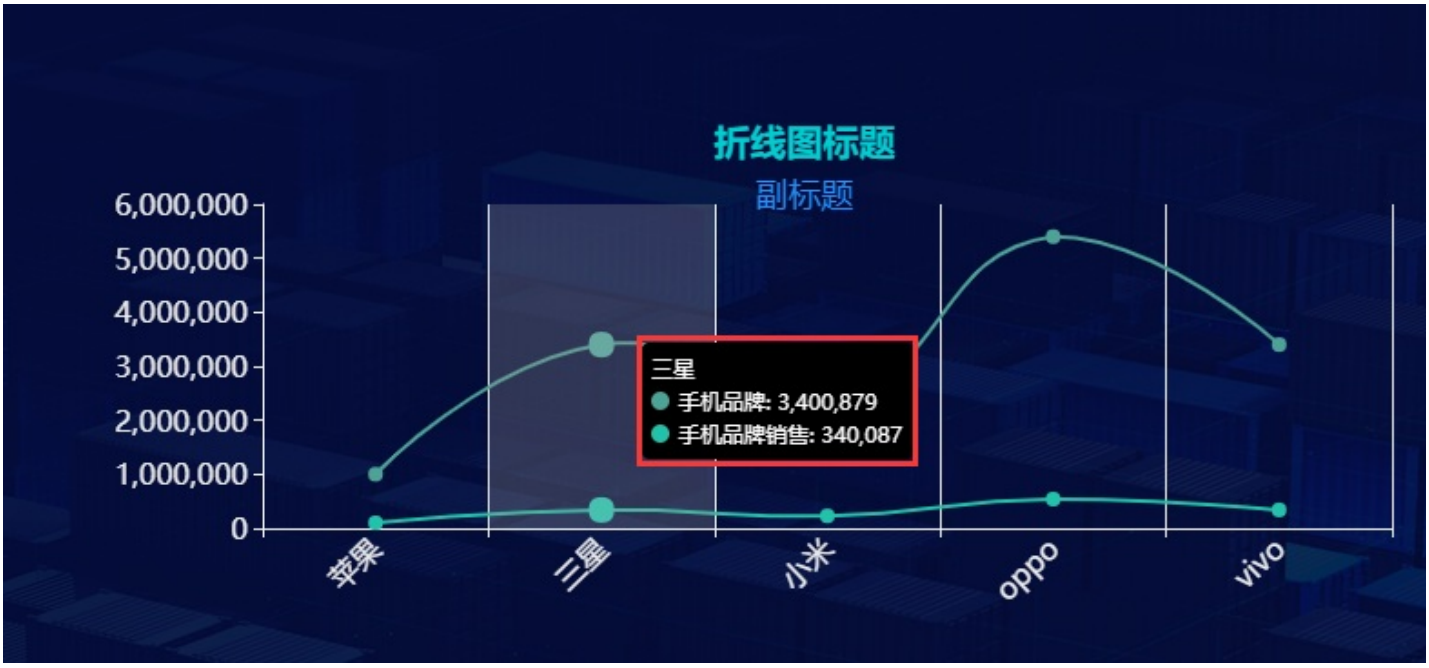


图2. 292

十、坐标轴边距设置

选中该折线图组件，在操作界面右侧的“坐标轴边距设置”处可修改折线图距离左、右、上、下的距离，如图2.293。

- 左边距（像素）：折线图距离左边的距离；
- 顶边距（像素）：折线图距离顶部的距离；

- 右边距（像素）：折线图距离右边的距离；
- 底边距（像素）：折线图距离底部的距离；

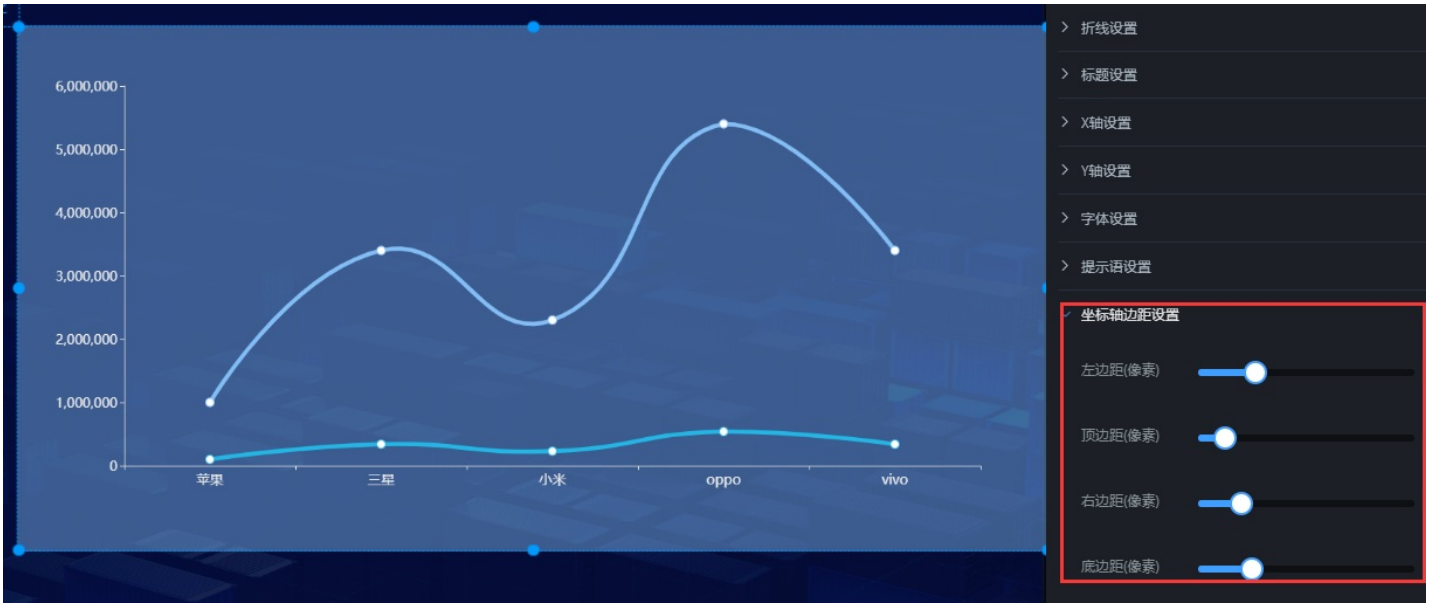


图2. 293

十一、图例设置

选中该折线图组件，在操作界面右侧的“图例设置”处可设置图例的样式，如图2.294；效果图如图2.295。

- 图例开关：是否显示图例；
- 字体颜色：图例的字体颜色。如果要想自定义图例的颜色，需要关闭系统配色（图2.2951）和删除所有自定义配色（图2.2952）中的颜色。
- 图例宽度：图例的宽度；
- 横向位置：图例的位置，分为：居中、左对齐、右对齐，如图2.2953；
- 纵向位置：图例的位置，分为：顶部、底部，如图2.2954；
- 布局朝向：图例的排列顺序，分为：横排和竖排，如图2.2955；
- 字体大小：图例的字体大小；


苹果
三星
小米
oppo
vivo
1000879
100087
3400879
340087
2300879
230087
5400879
540087
3400879
340087
Y轴设置
字体设置
提示语设置
坐标轴边距设置
图例操作
图例
位置
布局朝向
字体大小

图2.294

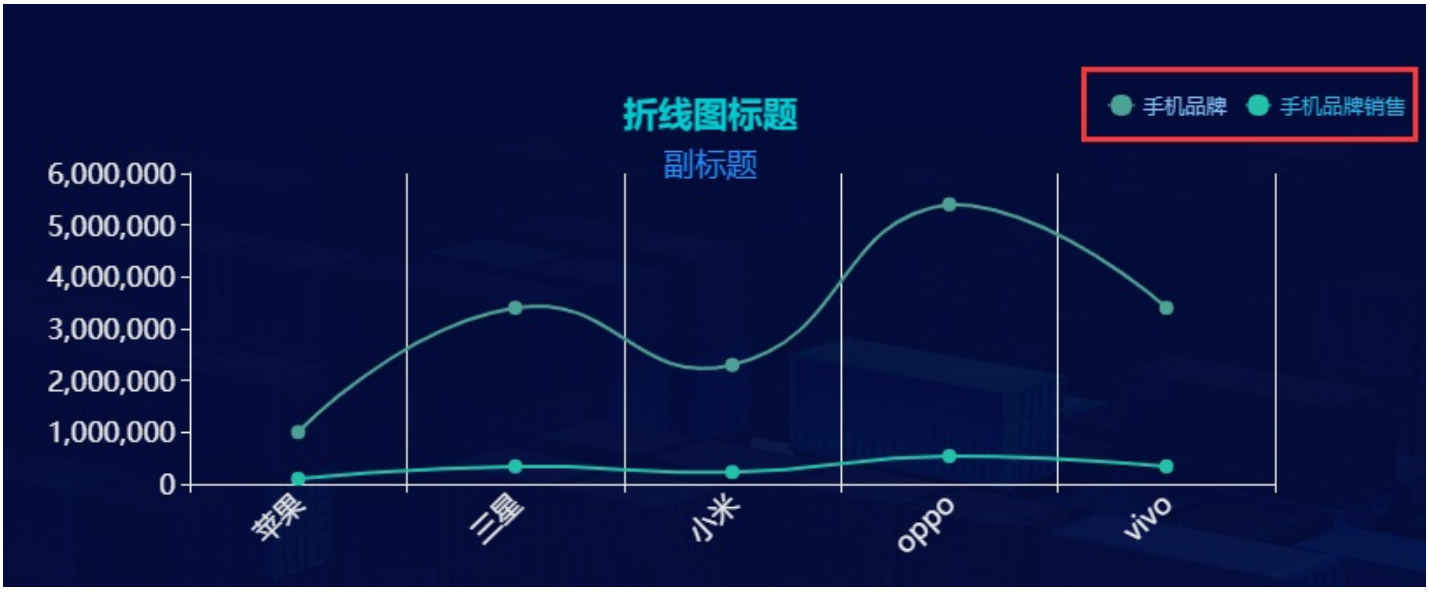

折线图标题
副标题
手机品牌
手机品牌销售
6,000,000
5,000,000
4,000,000
3,000,000
2,000,000
1,000,000
0
苹果
三星
小米
oppo
vivo

图2.295

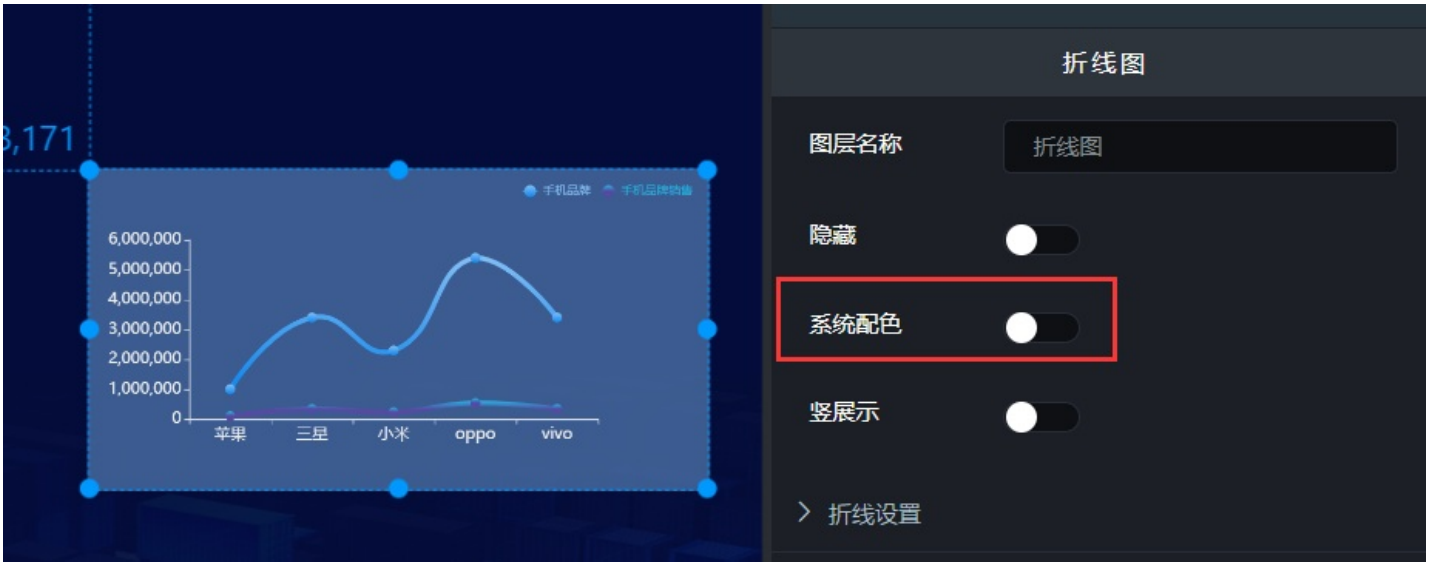

折线图
图层名称
折线图
隐藏
系统配色
竖展示
折线设置

图2.2951

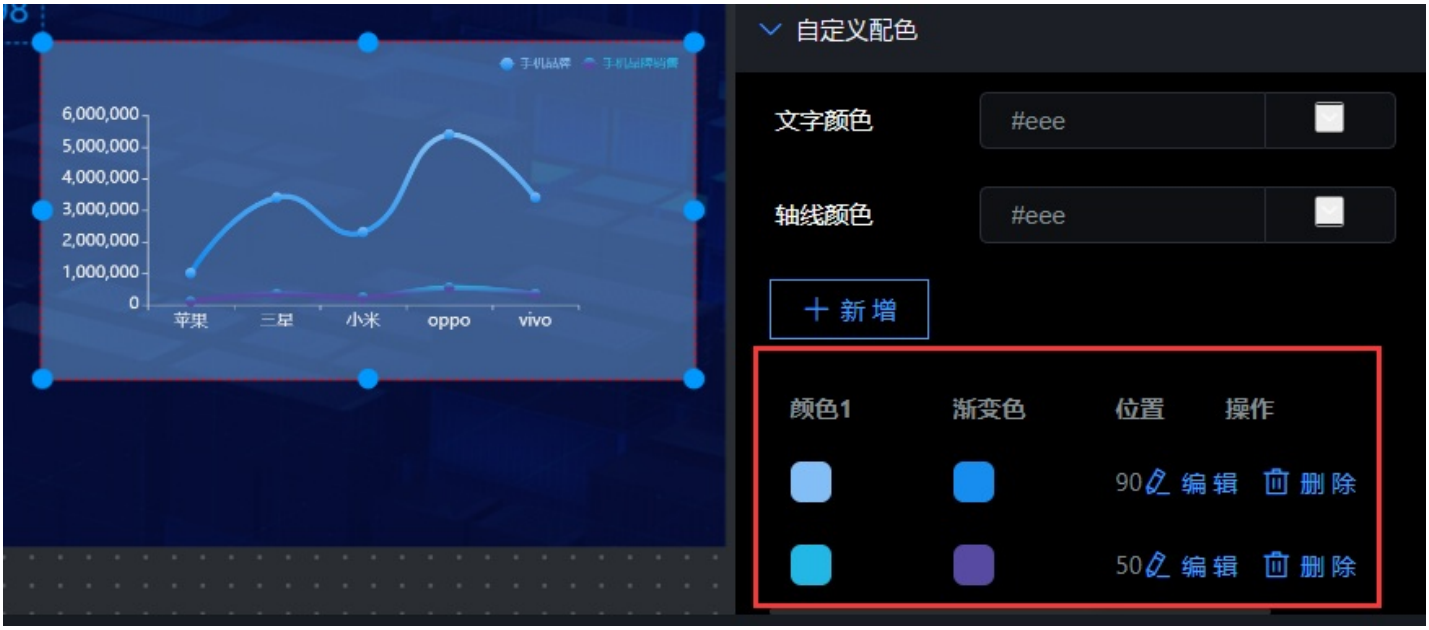
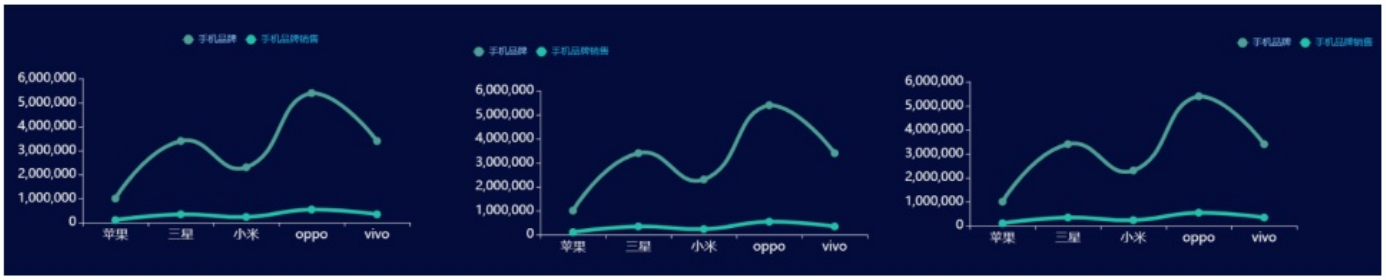
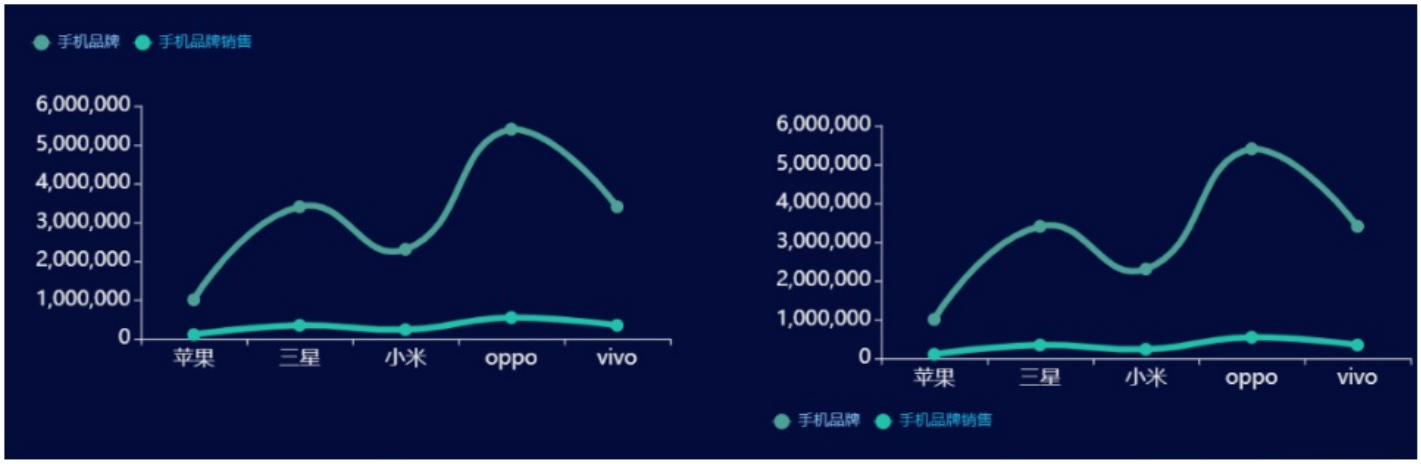


图2.2952



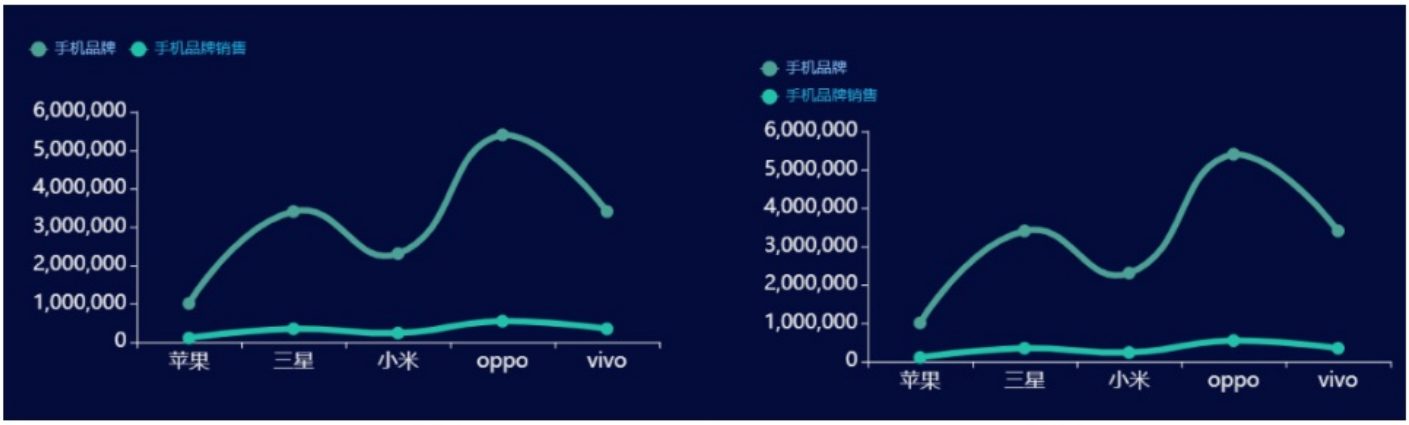
居中　　左对齐　　右对齐

图2.2953



顶部　　底部

图2.2954



竖排　　横排

图2.2955

十二、自定义配色设置

选中该折线图组件，在操作界面右侧的“自定义配色设置”处可配置上边不能设置的内容，如图2.296。

- 文字颜色：X、Y轴字体的颜色；
- 轴线颜色：X、Y轴轴线颜色；
- 配色：折线的颜色，如果开启了“系统配色”，需要先把系统配色先关掉，这样自定义的颜色才起作用；

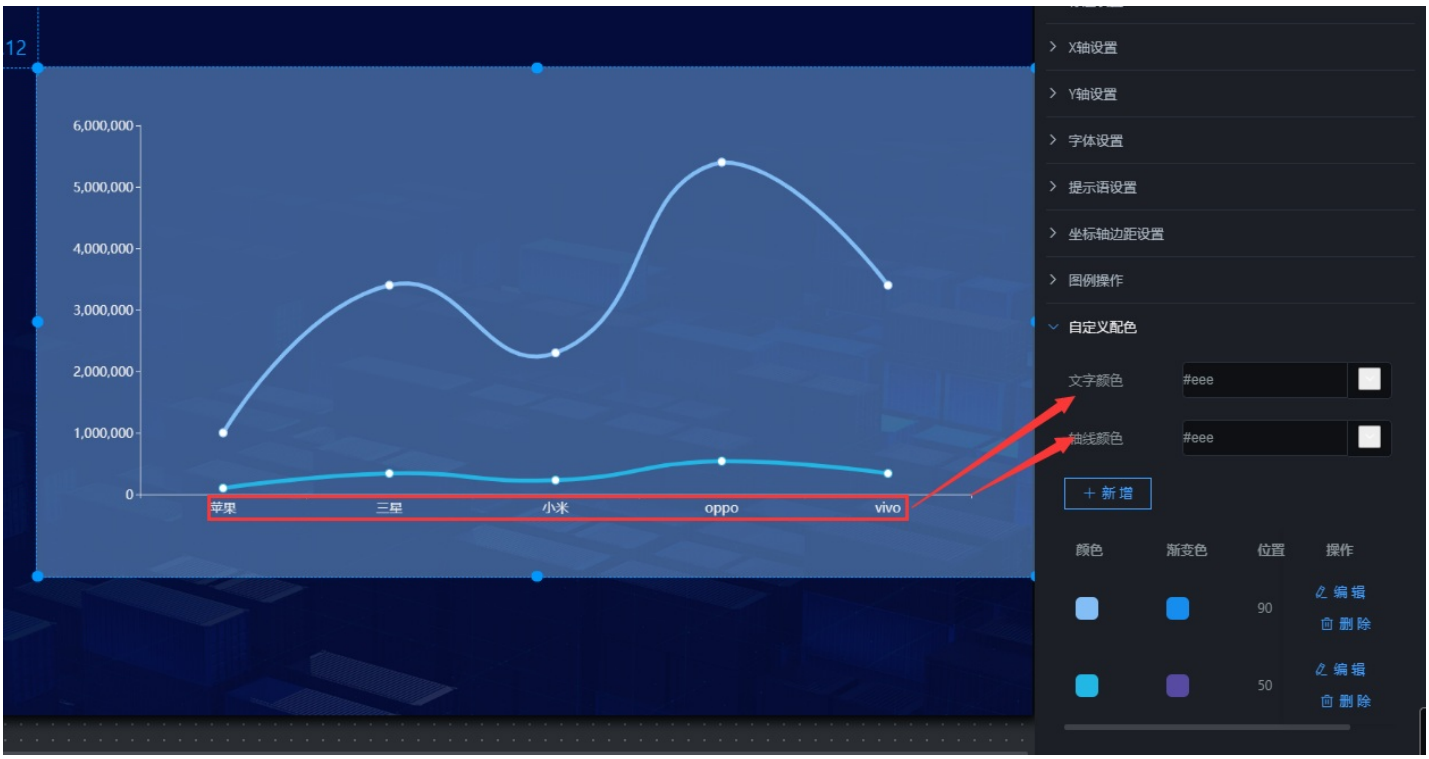


图2.296

十三、接口设置

选中该折线图组件，在操作界面右侧，点击“  ”，可设置接口，如图2.297。

1. 数据类型

数据类型分为静态数据和动态数据；

- 静态数据：写死的数据；
- 动态数据：会随着接口传过来的数据实时变化；（一般这种比较常用）

2. 接口地址

- categories：X轴参数；
- series：柱形内容；内部 name为柱形名称，data为柱形相对应Y轴的值；

（1）静态数据，接口地址传过来的内容要符合以下格式：

```
{"categories":["苹果","三星","小米","oppo","vivo"],"series":[{"name":"手机品牌","data":[1000879,3400879,2300879,5400879,3400879]},{"name":"手机品牌销售","data":[100087,340087,230087,540087,340087]}]}
```

（2）动态数据，接口地址传过来的内容要符合以下格式：

```
{"data":{"categories":["苹果","三星","小米","oppo","vivo"],"series":[{"name":"手机品牌","data":[1000879,3400879,2300879,5400879,3400879]},{"name":"手机品牌销售","data":[100087,340087,230087,540087,340087]}]}}
```

3. 刷新时间

这个参数主要针对动态数据设置的，完成数据的实时更新。

- 如果你想设置成5秒刷新一次，可以将刷新时间设置成“5000”；

4. 刷新数据

这个参数主要是重新请求以下接口，完成数据的更新。


数据类型
静态数据
动态数据
接口地址
接口方式
POST
GET
刷新时间
5000
数据处理

图2.297

2.3饼图组件

饼图组件就是添加饼图的组件。点击“  ”图标，再点击“饼图” ，即可创建新的图像，如图2.31；

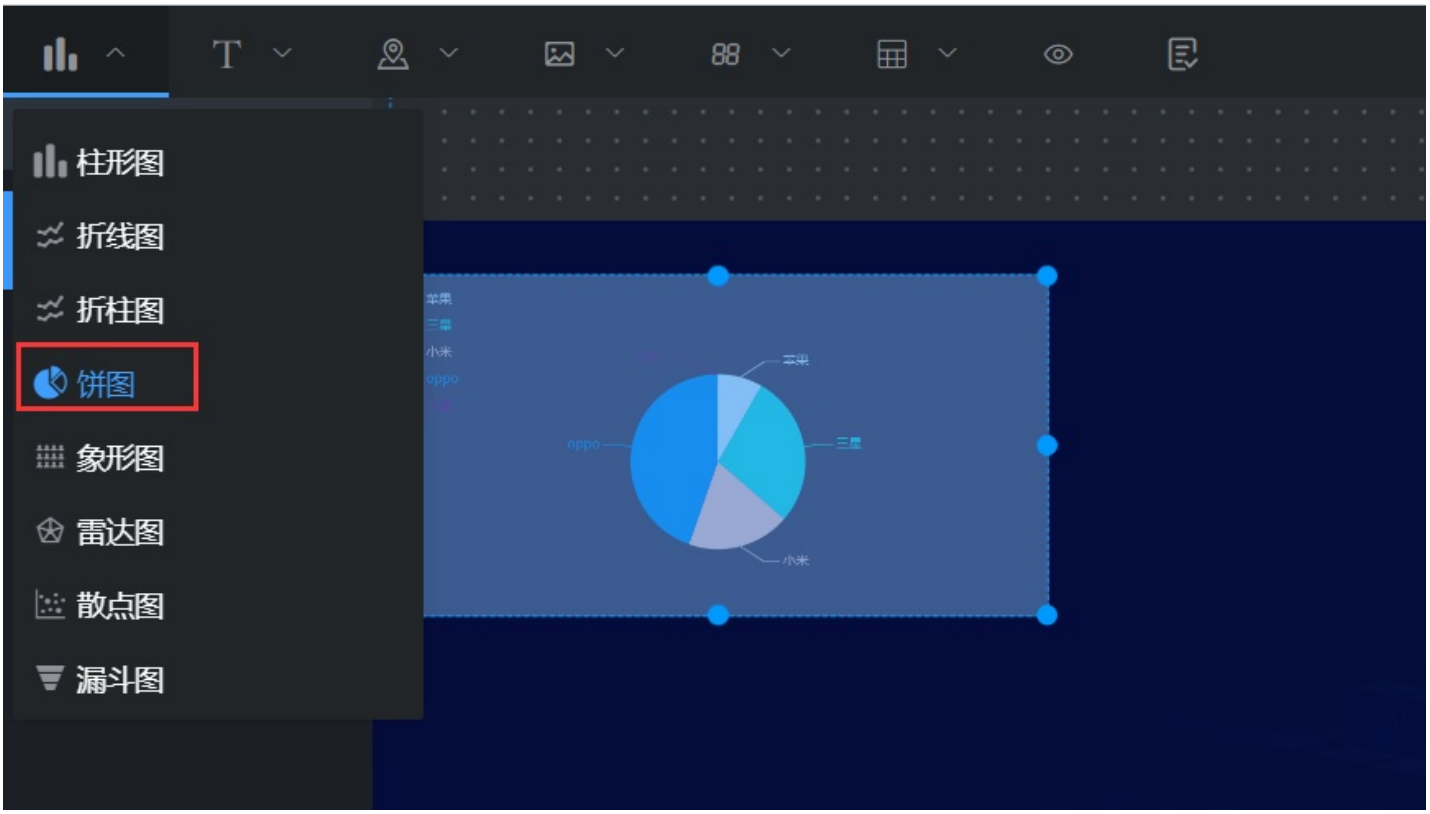


图2.31

一、组件名称设置

选中饼图组件，在操作界面右侧的“图层名称”处可修改组件的名称，如图2.32。（名称最好要设置一下，方便后期组件管理）

图2.32

二、系统配色

选中该饼图组件，在操作界面右侧，打开“系统配色”开关，在“配色选择”下拉框中选择主题，来设置饼图组件的配色，如图2.33。

- 默认配色：效果图如图2.331；
- 紫色主题：效果图如图2.332；
- 绿色主题：效果图如图2.333；

图2. 33

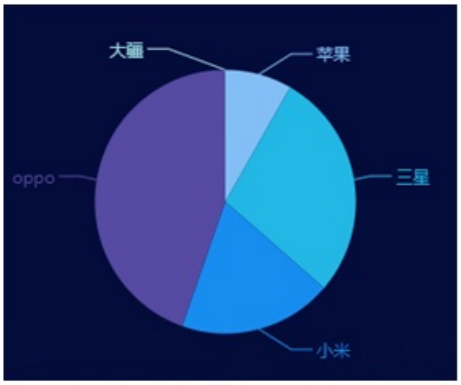


图 2.331

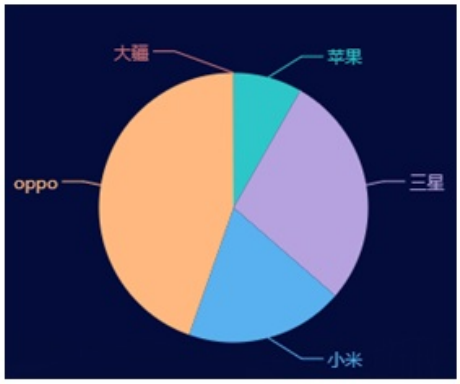


图 2.332

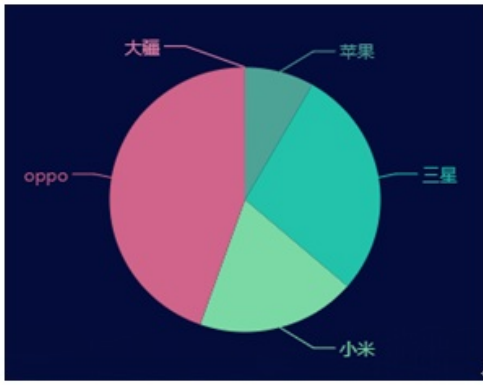


图 2.333

三、饼图设置

选中该饼图组件，在操作界面右侧的“饼图设置”处可修改设置组件的外观特点，如图2.34。

- 设为环形：把饼状图设置为环形样式；（关闭设为环形图开关，样式如图2.341；打开设为环形图开关，样式如图2.342）
- 半径：第一个百分比为内部圆的半径，如图2.342标注的1；第二个百分比为外部圆半径，如图2.342标注的2；（两个圆相差的部分为环）
- 南丁格尔玫瑰：把饼状图设置成玫瑰样式；（关闭南丁格尔玫瑰开关，样式如图2.341；打

开南丁格尔玫瑰开关，样式如图2.343）

- 自主排序：从小到大顺时针排序；
- 不显示零：不显示为0的份额；
- 标签位置：饼图上文字的位置，分为：外侧、内侧、中心；（标签位于外侧效果图如图2.344；标签位于内测，如图2.345；标签位于中心，效果图如图2.346）
- 链接：点击饼图跳转到的新页面，格式为”http://网址“或”https://网址“，如图2.347；

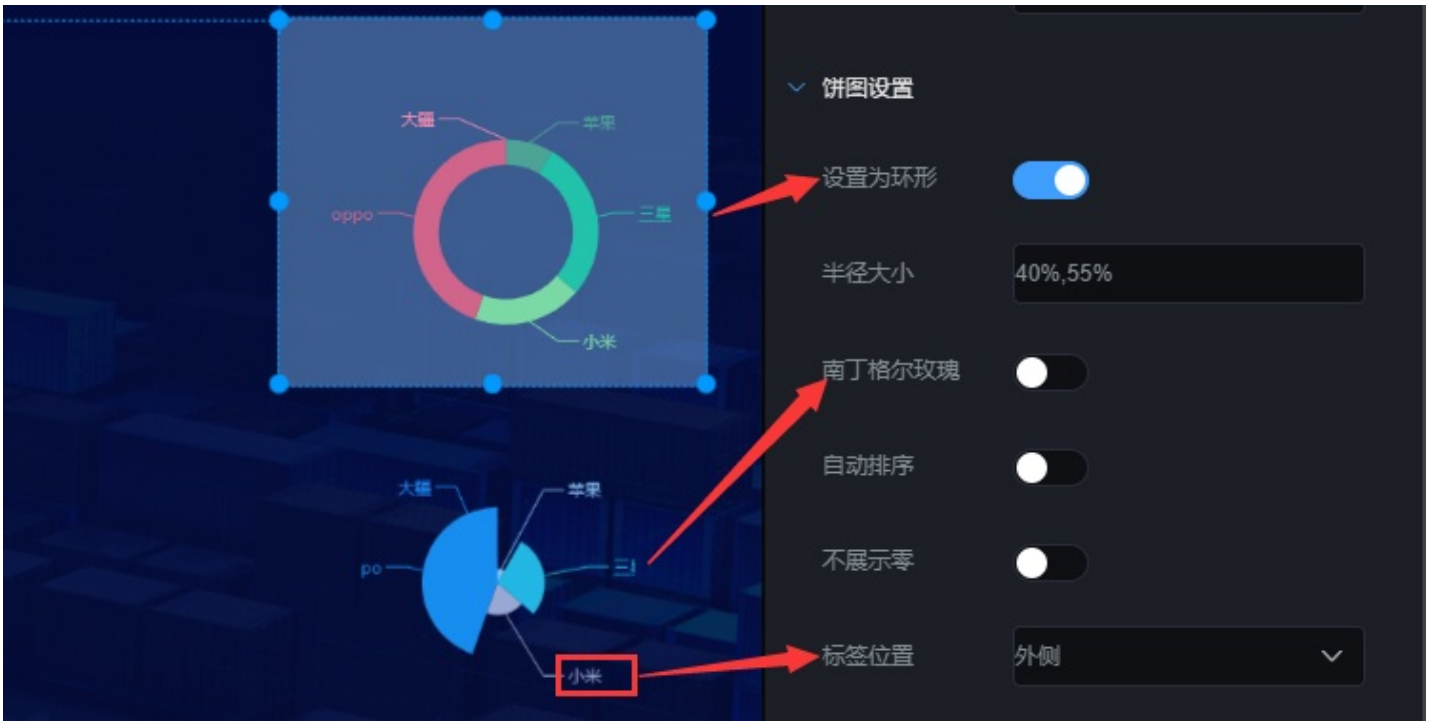


图2.34

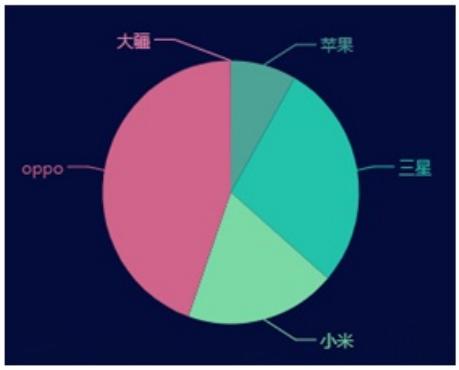


图 2.341

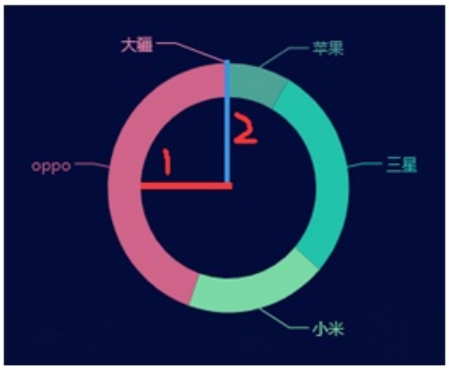


图 2.342

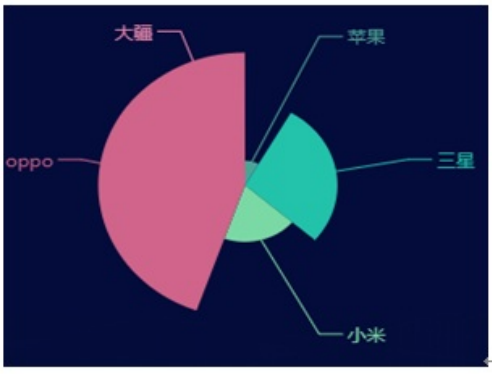


图 2.343

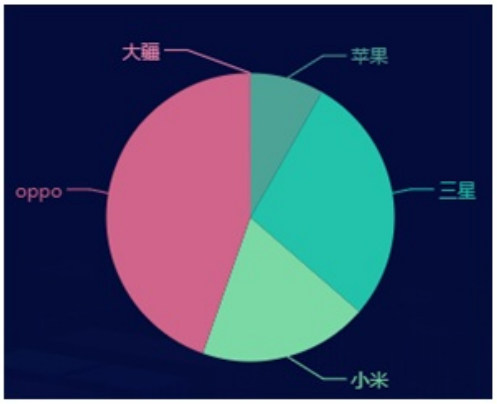


图 2.344

图 2.345

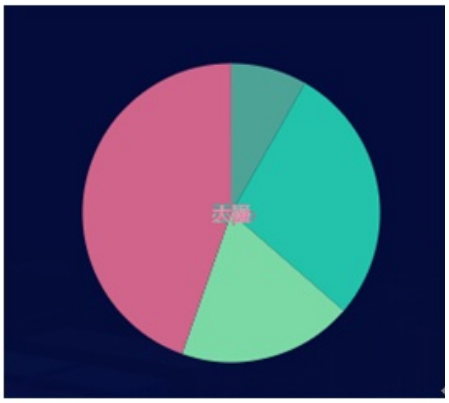


图 2.346

图2.347

四、标题设置

选中该饼图组件，在操作界面右侧的“标题设置”处可修改饼图组件的标题样式，如图2.35；效果图如图2.351。

- 标题开关：该开关控制标题的显示与隐藏；
- 标题：标题显示的内容；
- 字体颜色：标题的颜色；
- 字体大小：标题字体大小；
- 字体粗细：标题字体的粗细；
- 字体位置：标题的位置，分为：居中、左对齐、右对齐；
- 副标题：副标题内容；

- 字体颜色：副标题字体颜色；
- 字体粗细：副标题字体的粗细；
- 字体大小：副标题字体大小；

图2.35

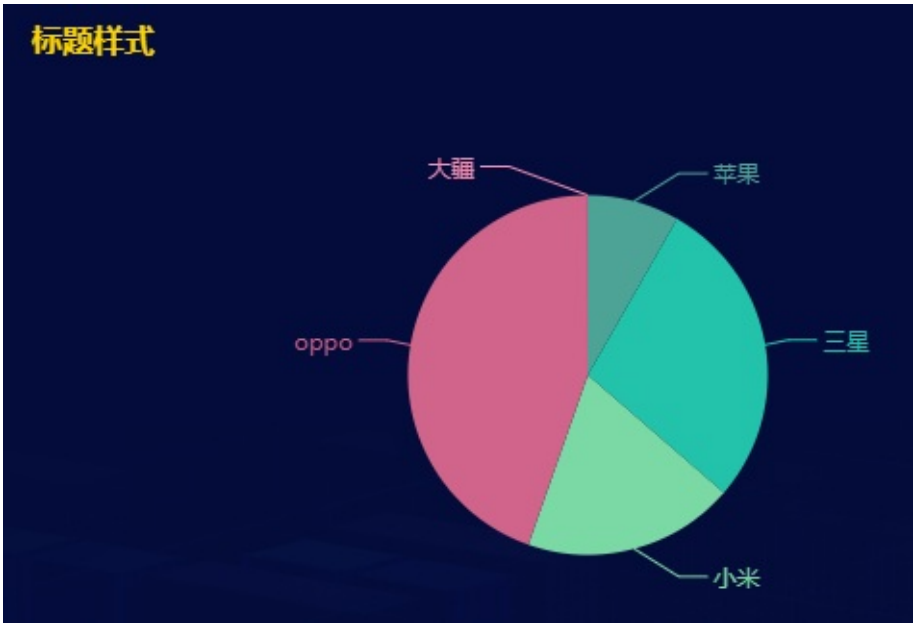


图2.351

五、字体设置

选中该饼图，在操作界面右侧的“字体设置”处可修改饼图组件标签的样式，如图2.36。

- 显示：标签是否显示；
- 字体大小：标签文字的大小；
- 字体颜色：标签文字的颜色；
- 字体粗细：标签文字的粗细；

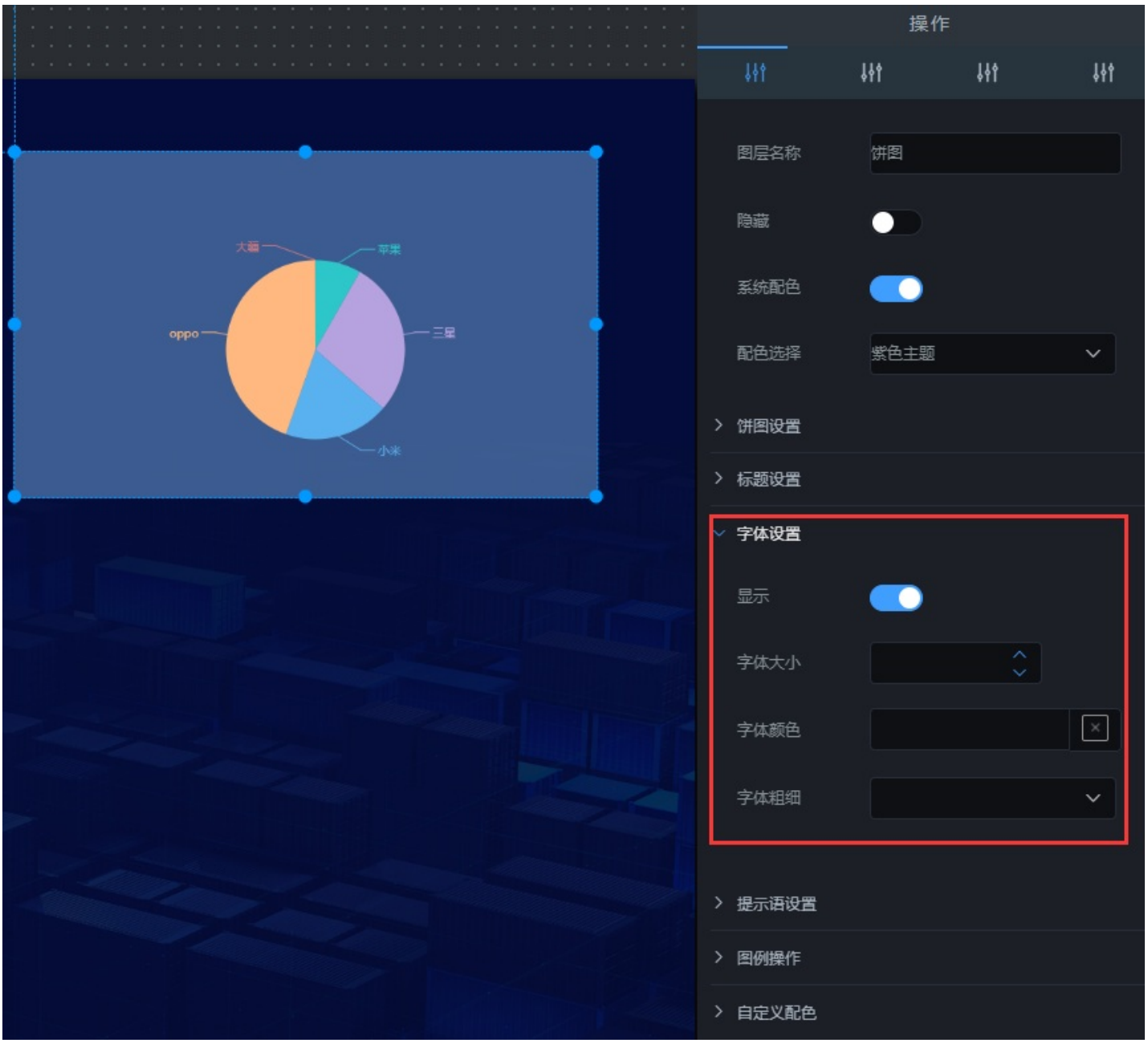


图2.36

六、提示语设置

选中该饼图组件，在操作界面右侧的“提示语设置”处可修改饼图组件的提示语，如图2.37。

- 字体大小：提示语的字体大小；
- 字体颜色：提示语的字体颜色；

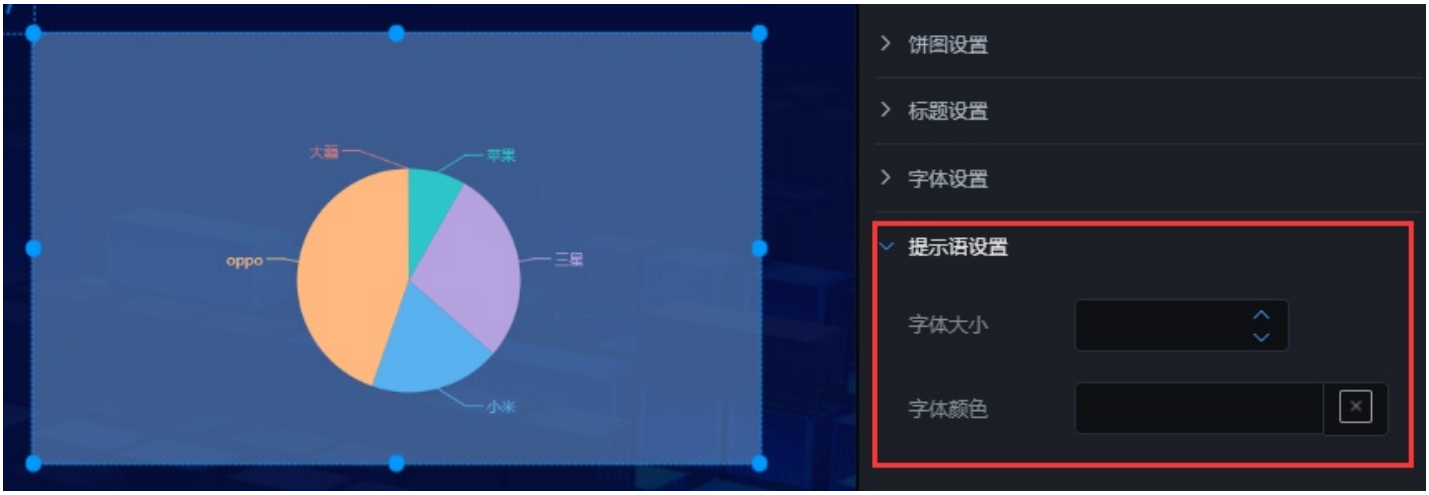


图2. 37

七、图例设置

选中该饼图组件，在操作界面右侧的“图例设置”处可设置图例的样式，如图2.38。

- 图例开关：是否显示图例；
- 字体颜色：图例的字体颜色。如果要想自定义图例的颜色，需要关闭系统配色（图2.381）和删除所有自定义配色（图2.382）中的颜色。
- 图例宽度：图例的宽度；
- 横向位置：图例的位置，分为：居中、左对齐、右对齐，如图2.383；
- 纵向位置：图例的位置，分为：顶部、底部，如图2.384；
- 布局朝向：图例的排列顺序，分为：横排和竖排，如图2.385；
- 字体大小：图例的字体大小；

图2. 38

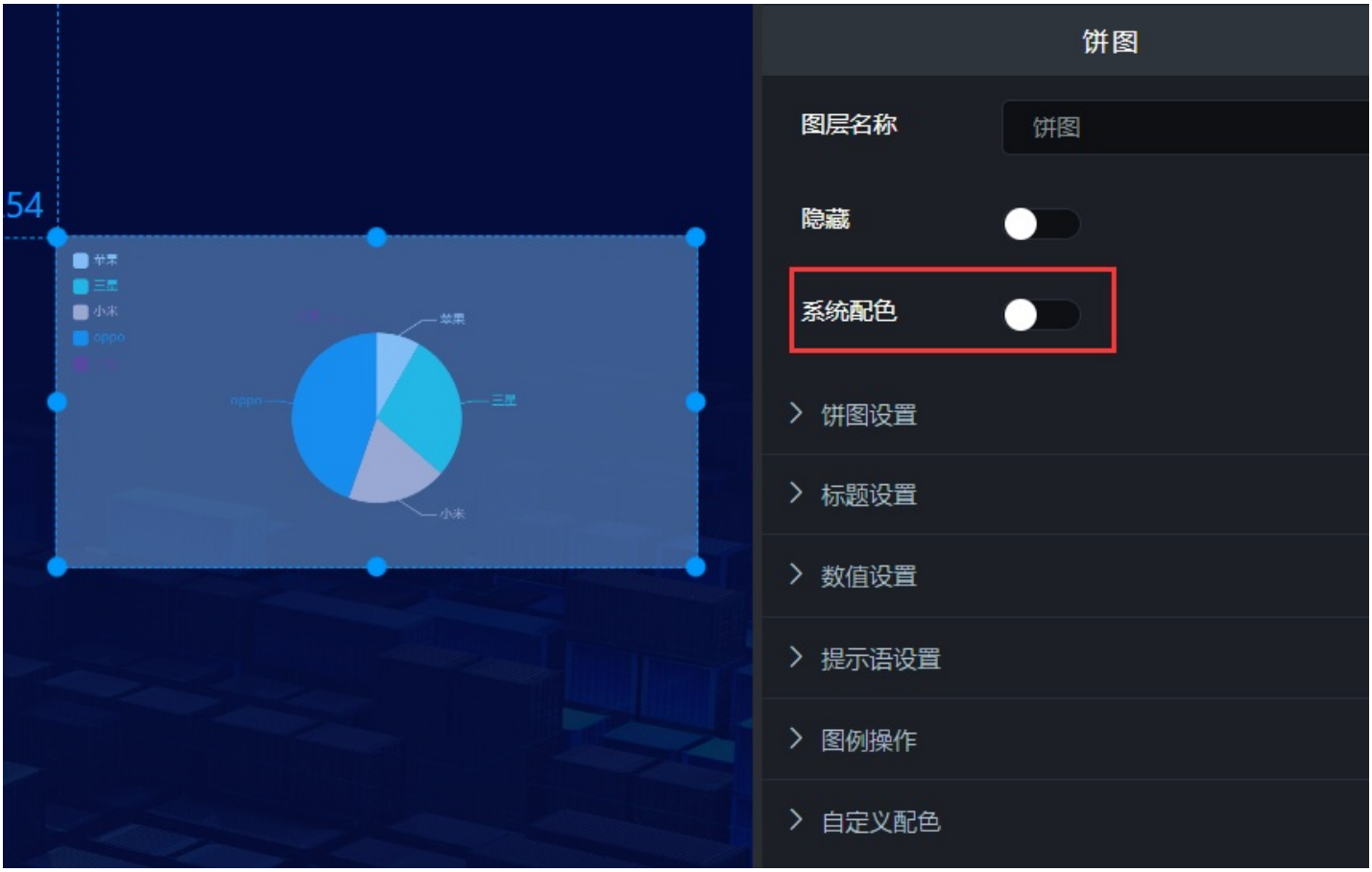
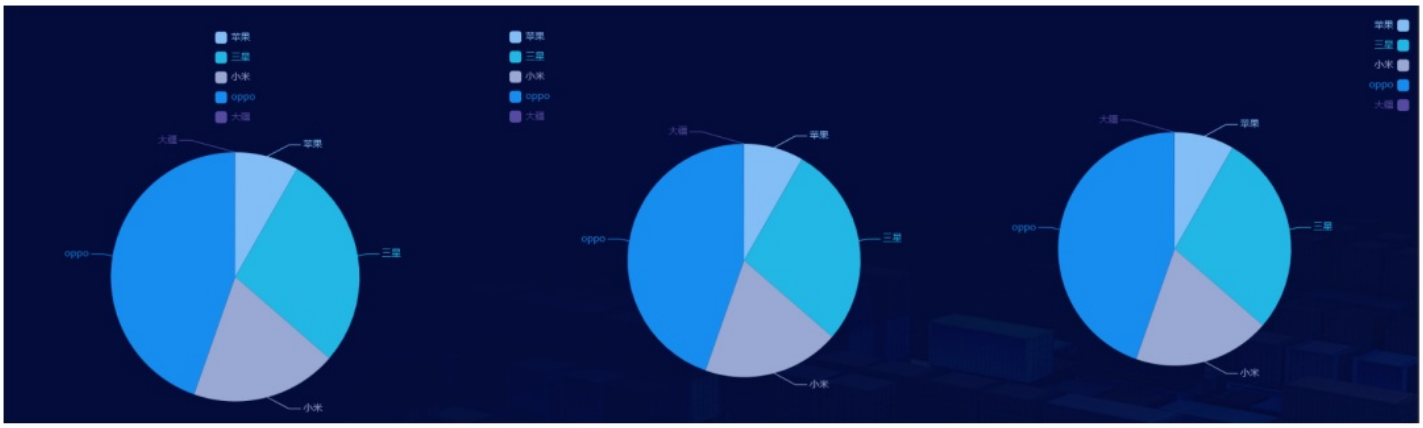

饼图
图层名称
饼图
隐藏
系统配色
饼图设置
标题设置
数值设置
提示语设置
图例操作
自定义配色
苹果
三星
小米
oppo

图2.381

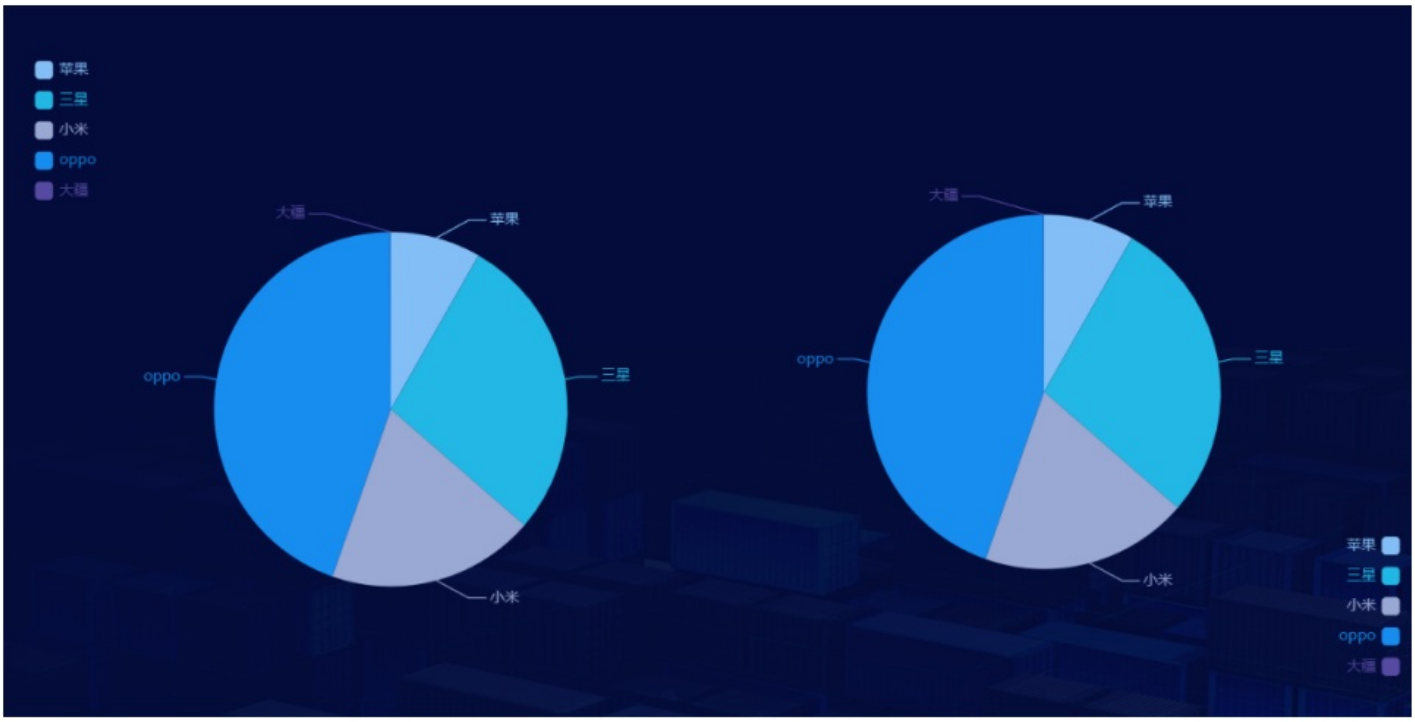

自定义配色
新增
颜色1
渐变色
位置
操作
编辑
删除
苹果
三星
小米
oppo

图2.382



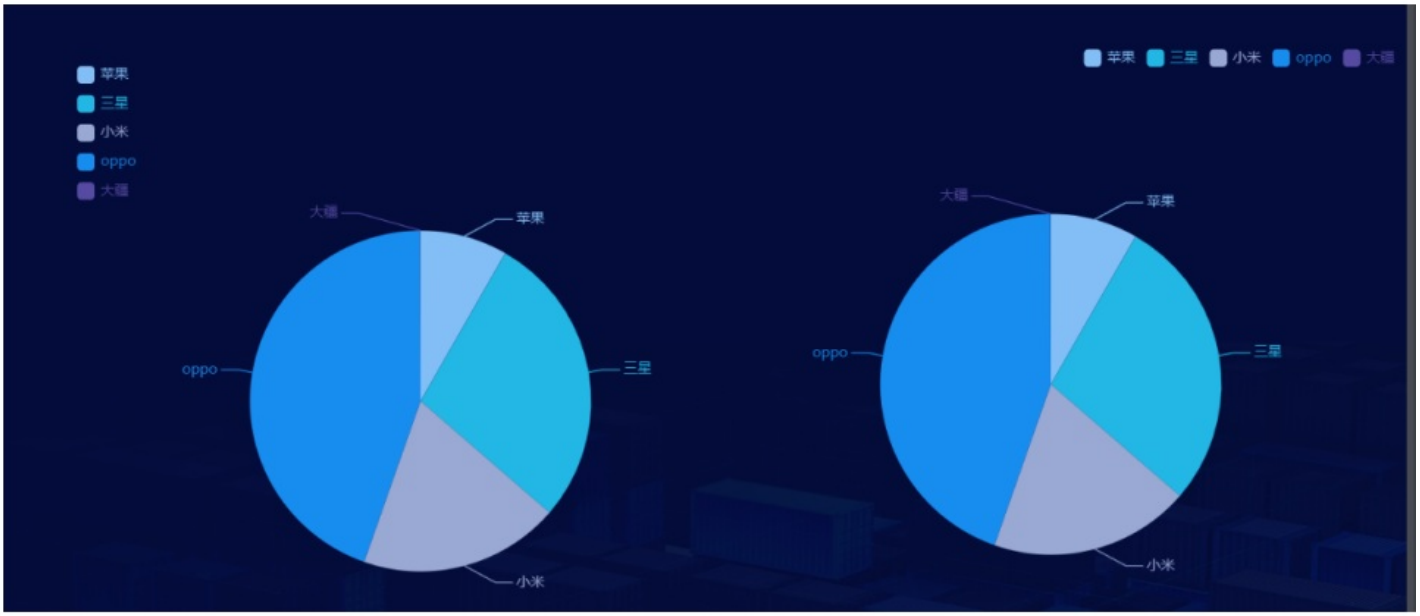
居中　　左对齐　　右对齐

图2. 383



顶部　　底部

图2. 384



竖排　　横排

图2.385

八、自定义配色设置

选中该饼图组件，在操作界面右侧的“自定义配色设置”处可配置上边不能设置的内容，如图2.39。

- 文字颜色：轴文字的颜色；（因为饼图没有轴，所以不需要设置）
- 轴线颜色：轴线颜色；（因为饼图没有轴，所以不需要设置）
- 配色：饼图的颜色，如果开启了“系统配色”，需要先把系统配色先关掉，这样自定义的颜色才起作用；

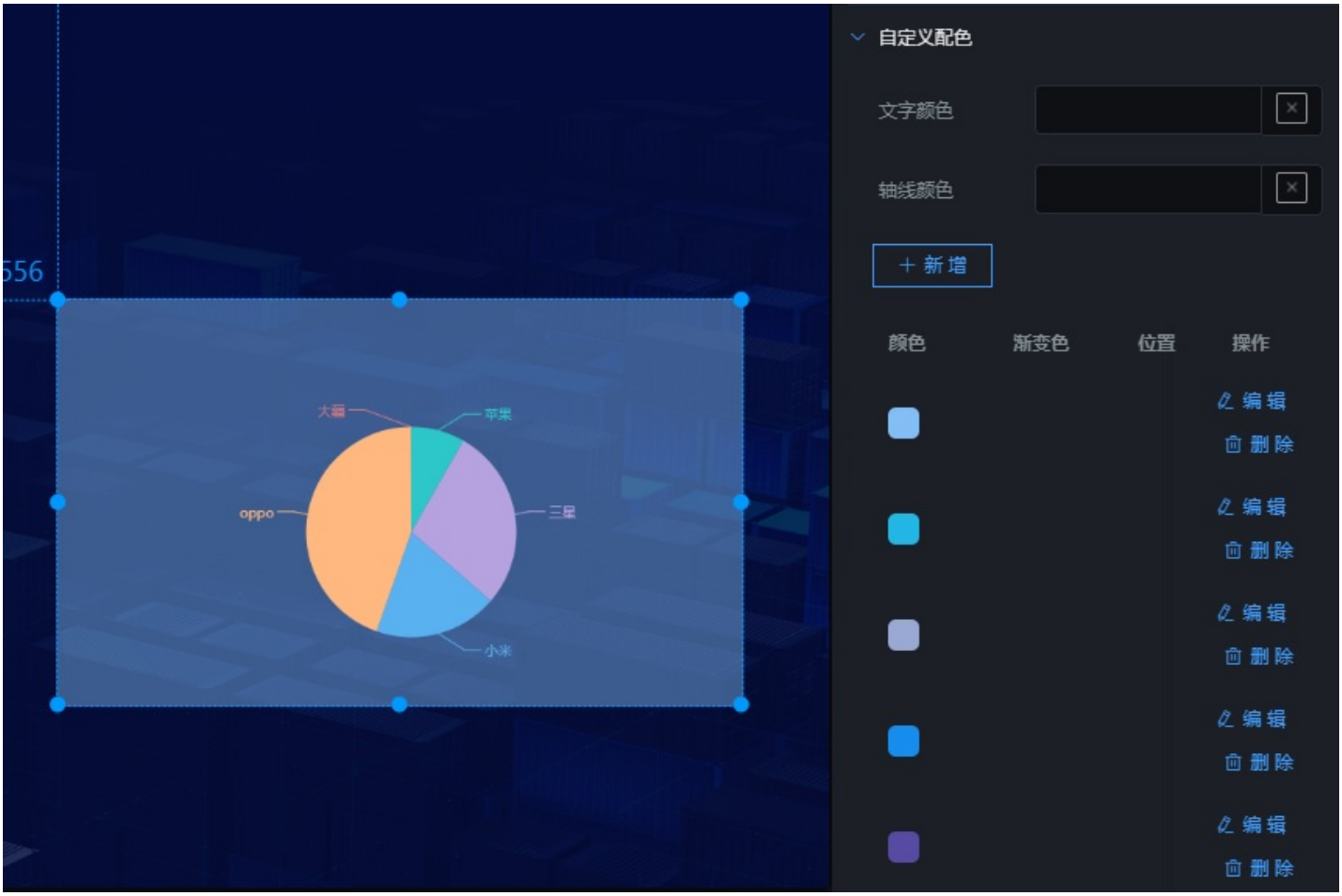


图2.39

九、接口设置

选中该饼图组件，在操作界面右侧，点击“  ”，可设置接口，如图2.391。

1. 数据类型

数据类型分为静态数据和动态数据；

- 静态数据：写死的数据；
- 动态数据：会随着接口传过来的数据实时变化；（一般这种比较常用）

2. 接口地址

- name：为标签名称；
- value：为标签值；

（1）静态数据，接口地址传过来的内容要符合以下格式：

```
[{"name":"苹果","value":1000879},{"name":"三星","value":3400879},{"name":"小米","value":2300879},{"name":"oppo","value":5400879},{"name":"大疆","value":3000}]
```

（2）动态数据，接口地址传过来的内容要符合以下格式：

```
{"data":[{"name":"苹果","value":1000879},{"name":"三星","value":3400879},{"name"
:"小米","value":2300879},{"name":"oppo","value":5400879},{"name":"大疆","value":
3000}]}
```

3. 刷新时间

这个参数主要针对动态数据设置的，完成数据的实时更新。

- 如果你想设置成5秒刷新一次，可以将刷新时间设置成“5000”；

4. 刷新数据

这个参数主要是重新请求以下接口，完成数据的更新。


数据类型
静态数据
动态数据
接口地址
接口方式
POST
GET
刷新时间
5000
数据处理

图2.391

2.4环形图组件

环形图组件就是环状显示某一事件进度的组件。点击“ ”图标，再点击“环形图”，即可创建环形图，如图2.41；

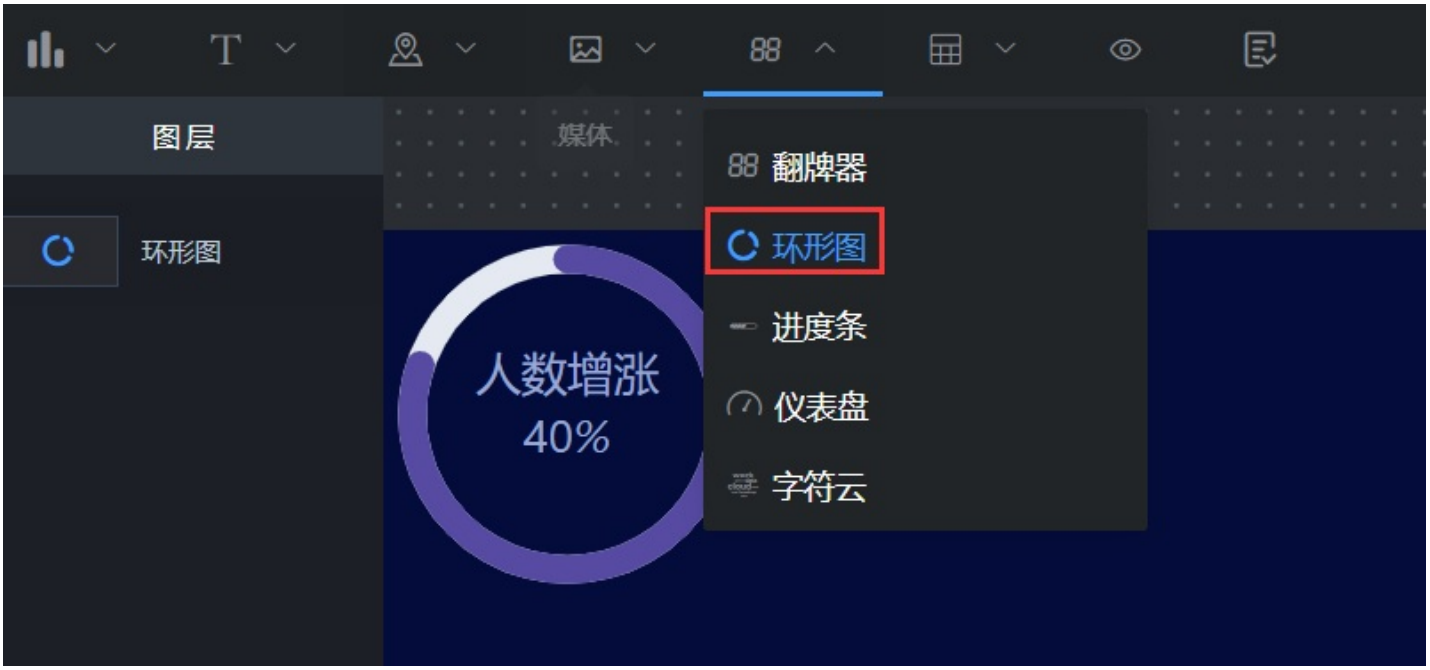


图2.41

一、组件名称设置

选中该环形图组件，在操作界面右侧的“图层名称”处可修改组件的名称，如图2.42。（名称最好要设置一下，方便后期组件管理）

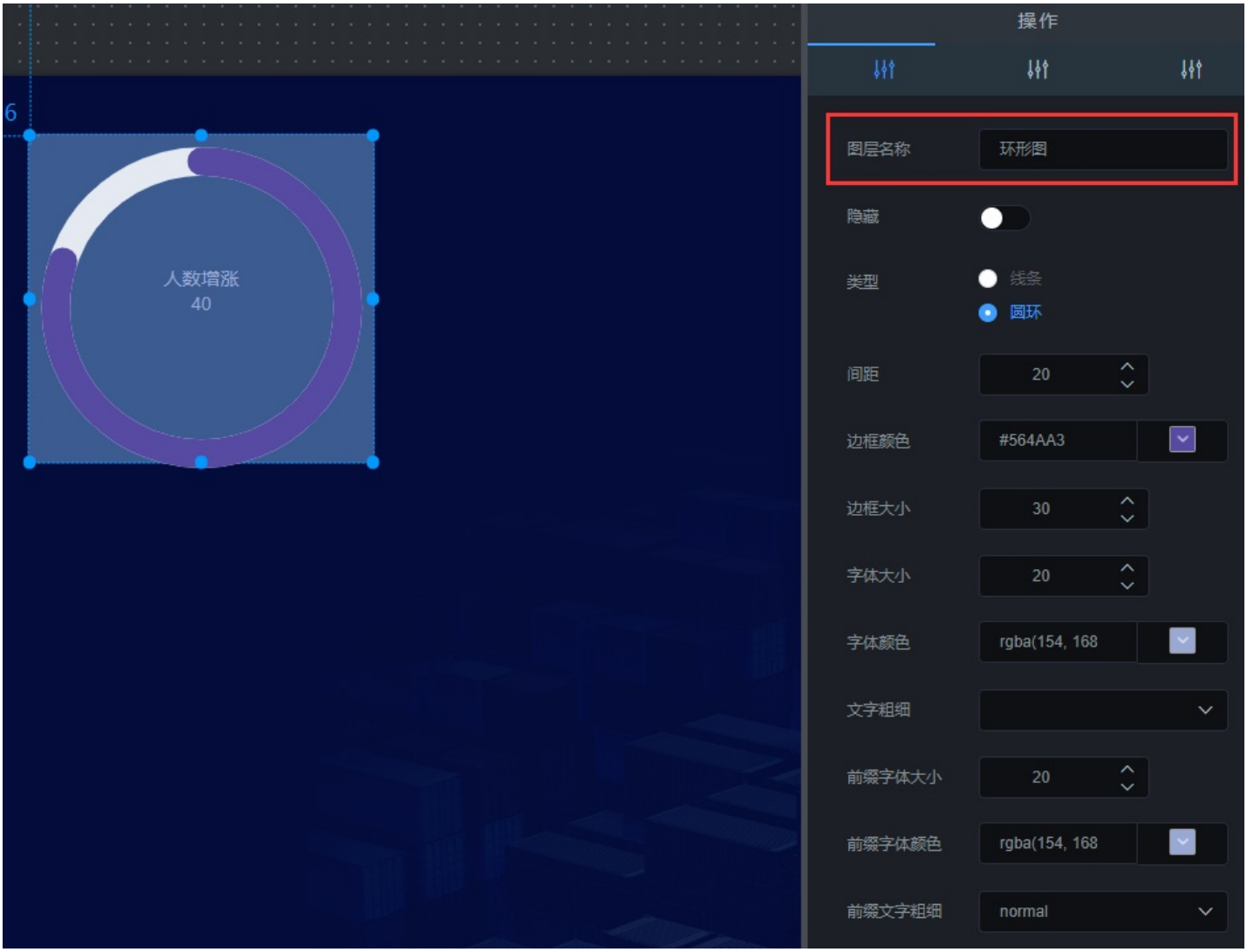


图2. 42

二、类型

选中该环形图组件，在操作界面右侧的“类型”处可选择显示样式，如图2.43；类型包含：线条和圆环。

- 条线图：进度条样式；效果图如图2.431。
- 环形图：环状样式；效果图如图2.432。

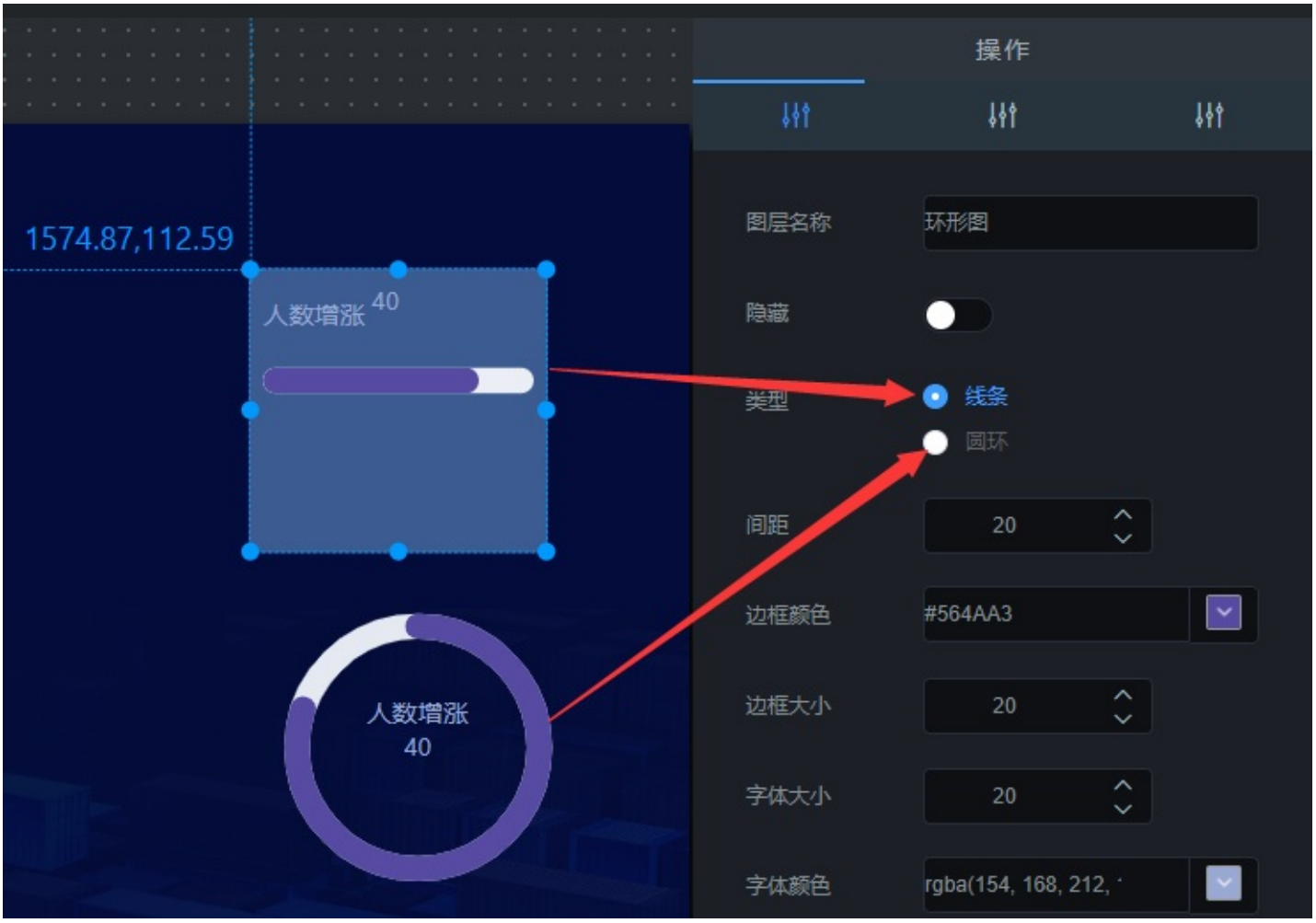


图2. 43

图 2.431

图 2.432

三、 间距

选中该环形图组件，在操作界面右侧的“间距”处可修改文字和进度条的间距，如图2.44。

备注：数值越大，文字离进度条越远

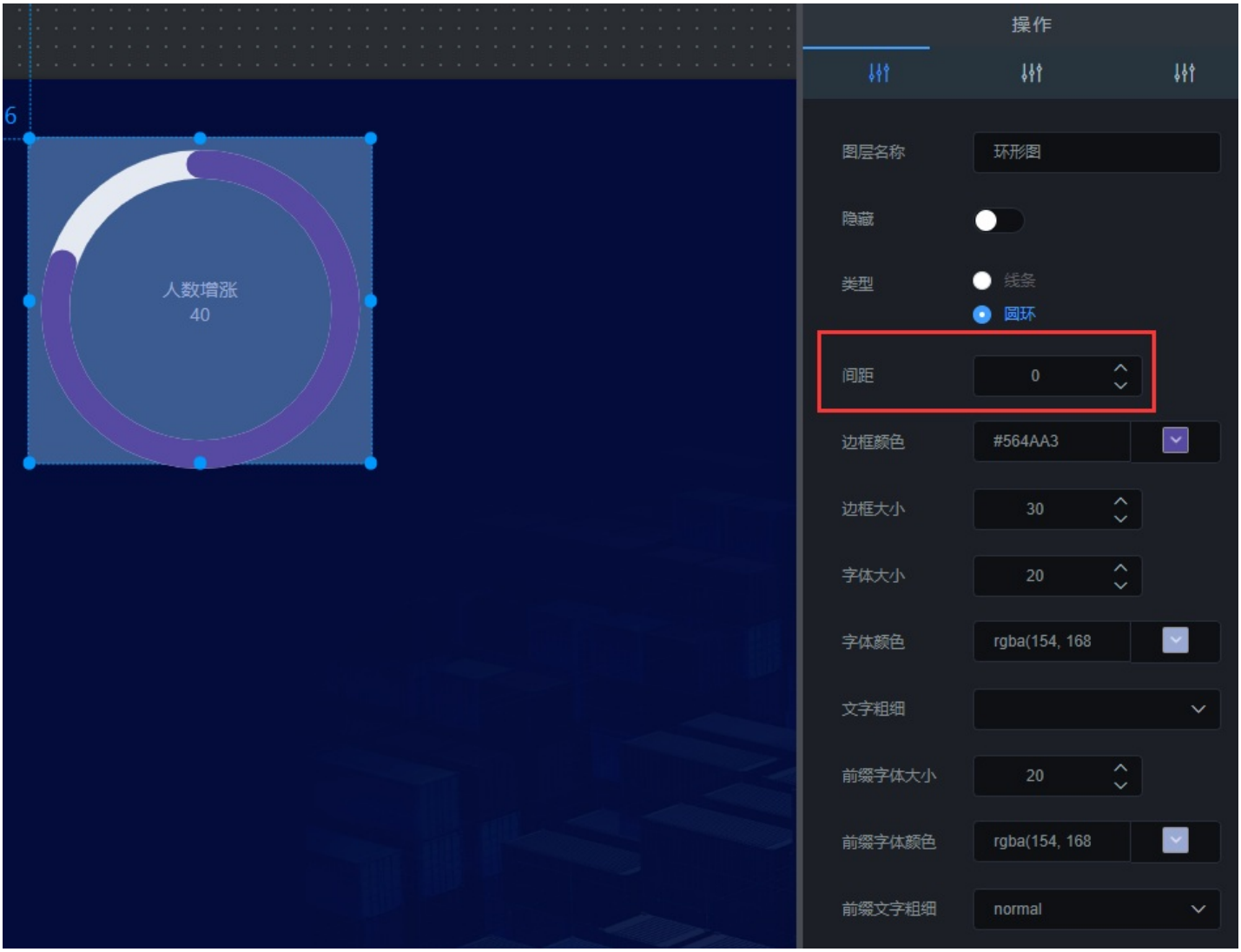


图2. 44

四、边框颜色

选中该环形图组件，在操作界面右侧的“边框颜色”处可修改进度条的颜色，如图2.45。

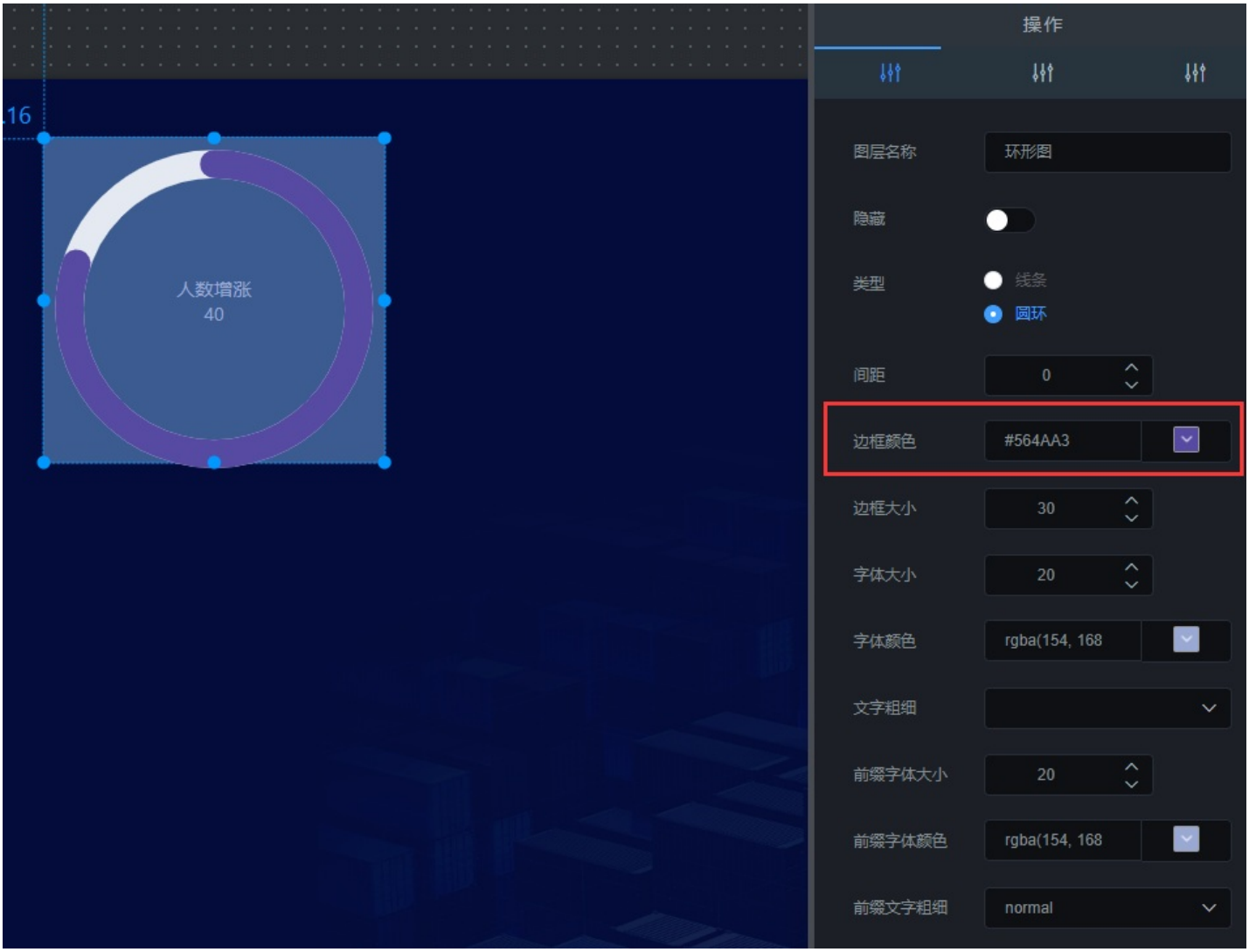


图2.45

五、字体设置

字体大小：修改图2.46标注文字大小；

字体颜色：修改图2.46标注文字颜色；

文字粗细：修改图2.46标注文字粗细；

图2. 46

六、前缀设置

前缀字体大小：修改图2.47标注文字大小；

前缀字体颜色：修改图2.47标注文字颜色；

前缀文字粗细：修改图2.47标注文字粗细；

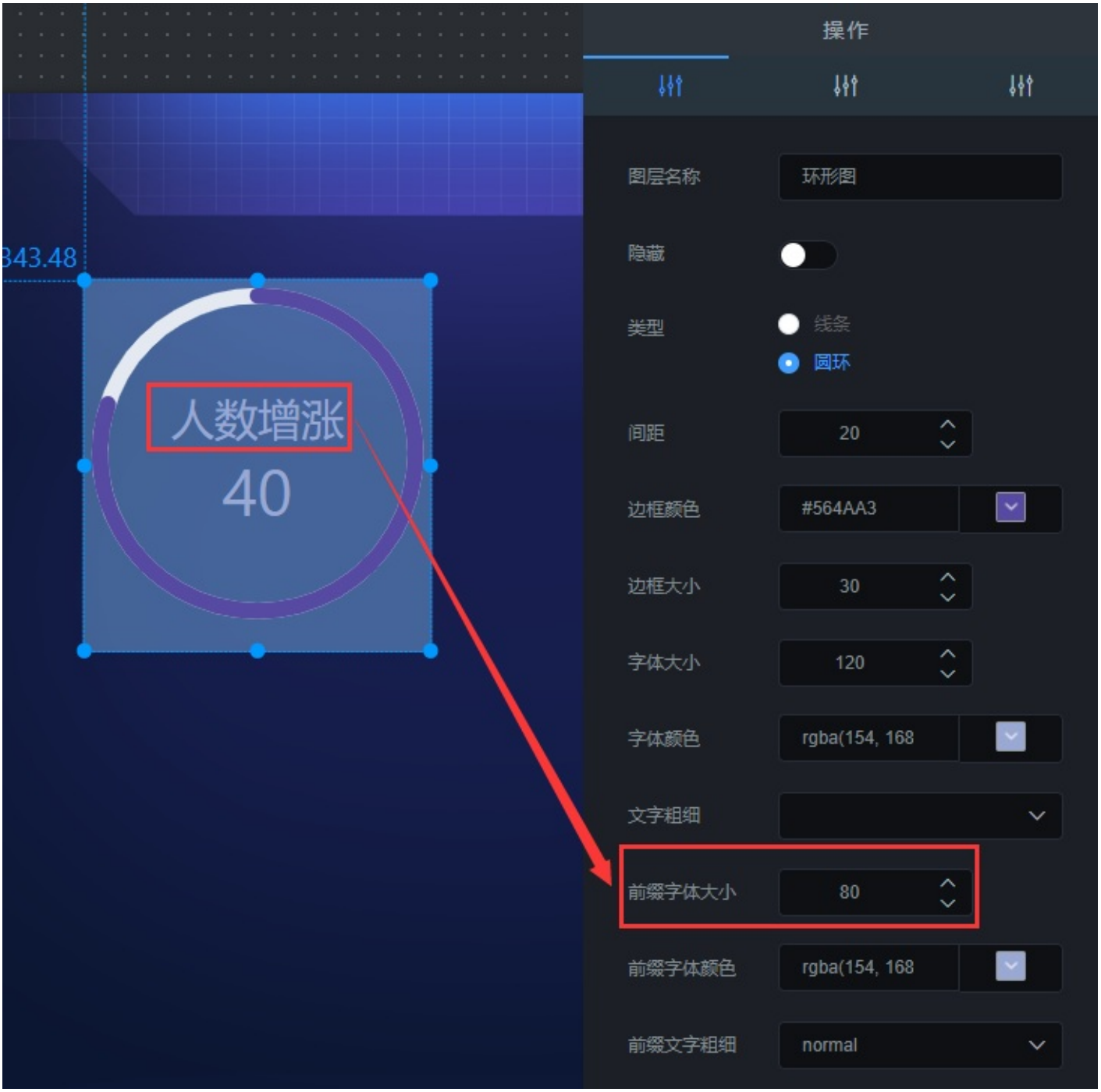


图2. 47

七、接口设置

1. 数据类型

数据类型分为静态数据和动态数据；

- 静态数据：写死的数据；
- 动态数据：会随着接口传过来的数据实时变化；（一般这种比较常用）

2. 接口地址

- label：为前缀内容；

- value：为文字内容；
- data：为百分比；

（1）静态数据，接口地址穿过来的内容要符合以下格式：

```
{"label":"人数增涨","value":40,"data":90}
```

（2）动态数据，接口地址穿过来的内容要符合以下格式：

```
{"data":{"label":"人数增涨","value":40,"data":90}}
```

3. 刷新时间

这个参数主要针对动态数据设置的，完成数据的实时更新。

- 如果你想设置成5秒刷新一次，可以将刷新时间设置成“5000”；

4. 刷新数据

这个参数主要是重新请求以下接口，完成数据的更新。

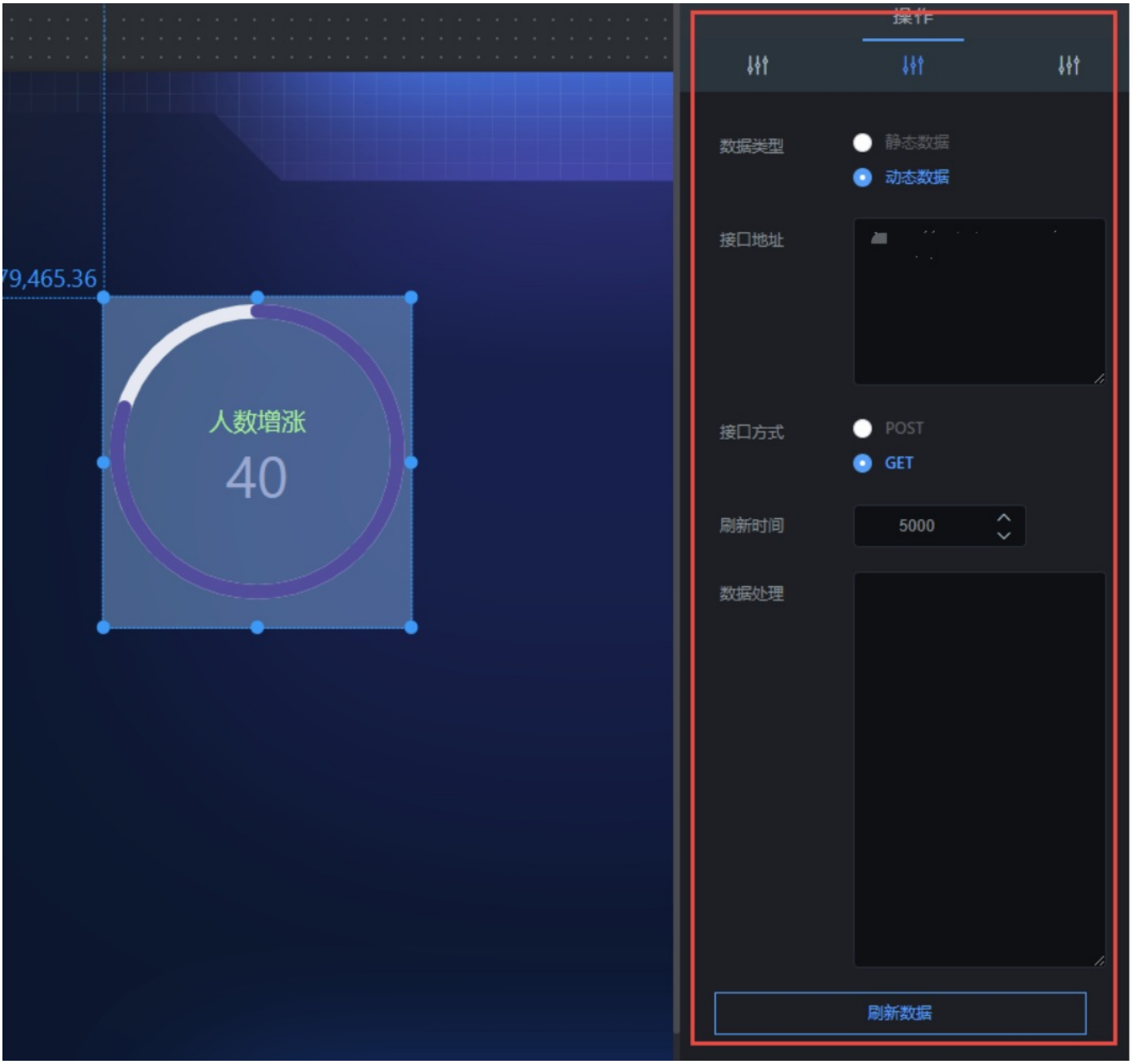


图2.48

5.其他样式，怎样设置**

如图2.49中样式，接口设置格式如下：

图2.49

备注：如果接口中配置“content”和“data”值，环形图就显示百分值；

（1）静态数据，接口地址穿过来的内容要符合以下格式：

```
{ "label": "人数增涨",
 "value": 40,
 "content": "%", "data": 80}
```

（2）动态数据，接口地址穿过来的内容要符合以下格式：

```
{"data":{ "label": "人数增涨",
 "value": 40,
 "content": "%", "data": 80}}
```

2.5象形图组件

象形图组件就是添加象形图的组件。点击“  ”图标，再点击“象形图”，即可创建新的图像，如图2.51；

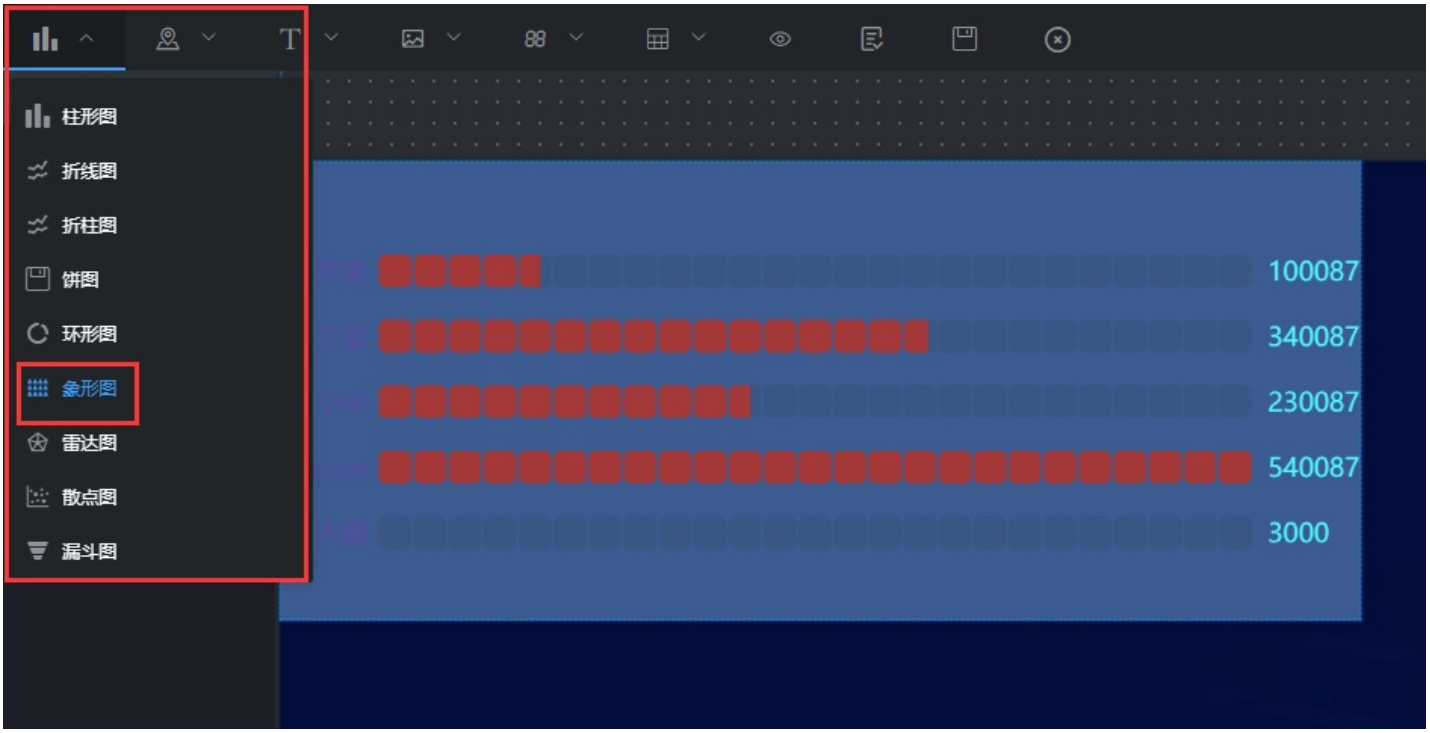


图2.51

一、组件名称设置

选中象形图组件，在操作界面右侧的“图层名称”处可修改组件的名称，如图2.52。（名称最好要设置一下，方便后期组件管理）

图2.52

二、图标

选中该象形组件，在操作界面右侧，输入图片地址，即可自定义设置象形图图标，如图2.53。

图2.53

三、图标字号

选中该象形图组件，在操作界面右侧，选择图标字号，可设置图标的大小，如图2.54。

图2.54

四、字体设置

包含设置“字体大小、字体颜色”，如图2.55。

图2.55

五、轴字体颜色

选中该象形图组件，在操作界面右侧，选择轴字体颜色，可设置轴文字的颜色（即Y轴文字的颜色），如图2.56。

图2.56

六、间距

选中该象形图组件，在操作界面右侧，调整“间距”大小，可修改组件Y轴之间的间距，如图2.57。

图2.57

七、Y轴设置

选中该象形图组件，在操作界面右侧的“Y轴设置”处可设置Y轴的样式，如图2.58。

- 显示：是否显示Y轴字体；
- 字号：Y轴字体的大小；

图2.58

八、坐标轴边距设置

选中该象形图组件，在操作界面右侧的“坐标轴边距设置”处可配置距离左、右、上、下的距离，如图2.59。

- 左边距（像素）：柱形图距离左边的距离；
- 顶边距（像素）：柱形图距离顶部的距离；
- 右边距（像素）：柱形图距离右边的距离；
- 底边距（像素）：柱形图距离底部的距离；

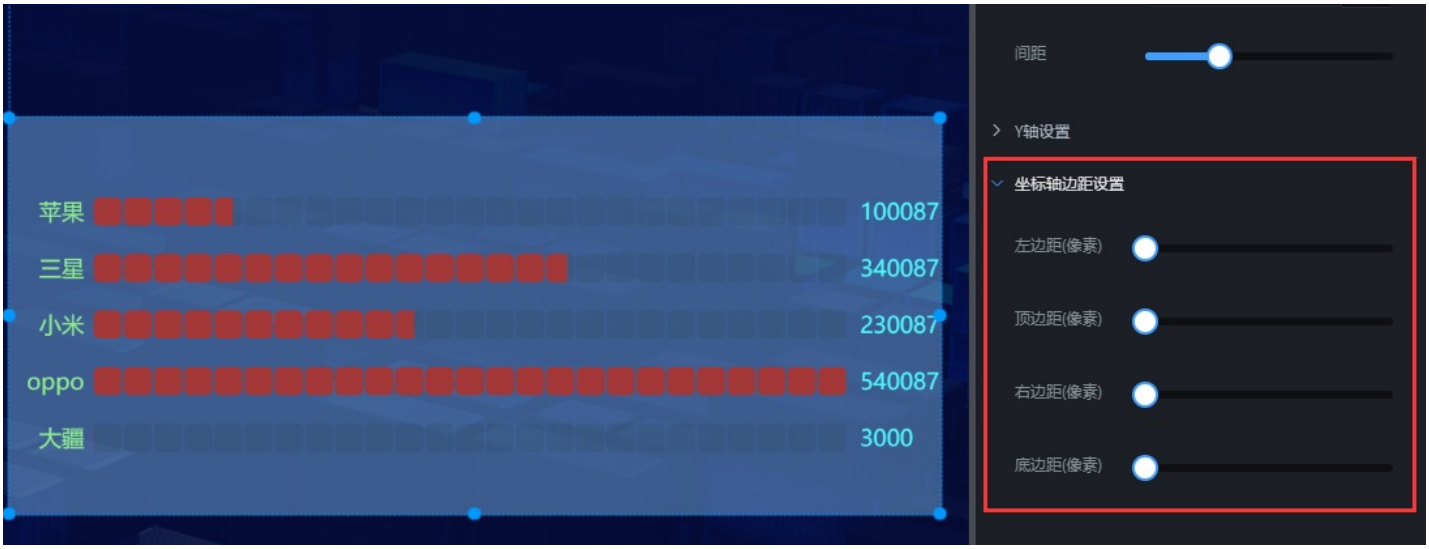


图2.59

九、接口设置

选中该象形图组件，在操作界面右侧，点击“  ”，可设置接口，如图2.591。

1. 数据类型

数据类型分为静态数据和动态数据；

- 静态数据：写死的数据；
- 动态数据：会随着接口传过来的数据实时变化；（一般这种比较常用）

2. 接口地址

- name：为标签名称；
- value：为标签值；

（1）静态数据，接口地址传过来的内容要符合以下格式：

```
[
    {
        "name": "苹果",
        "value": 1000879
    },
    {
        "name": "三星",
        "value": 3400879
    },
    {
        "name": "小米",
        "value": 2300879
    },
    {
        "name": "oppo",
        "value": 5400879
    },
    {
        "name": "大疆",
        "value": 3000
    },
    {
        "name": "抖音",
        "value": 2000
    }
]
```

（2）动态数据，接口地址传过来的内容要符合以下格式：

```
{"data":[{"name":"苹果","value":1000879},{"name":"三星","value":3400879},{"name"
:"小米","value":2300879},{"name":"oppo","value":5400879},{"name":"大疆","value":
3000}]}
```

3. 刷新时间

这个参数主要针对动态数据设置的，完成数据的实时更新。

- 如果你想设置成5秒刷新一次，可以将刷新时间设置成“5000”；

4. 刷新数据

这个参数主要是重新请求以下接口，完成数据的更新。


数据类型
静态数据
动态数据
接口地址
k/26/bar
接口方式
POST
GET
刷新时间
5000
数据处理

图2.591

2.6雷达图组件

雷达图组件就是添加雷达图的组件。点击“ ”图标，再点击“雷达图”，即可创建新的图像，如图2.61；

图2.61

一、组件名称设置

选中雷达图组件，在操作界面右侧的“图层名称”处可修改组件的名称，如图2.62。（名称最好要设置一下，方便后期组件管理）

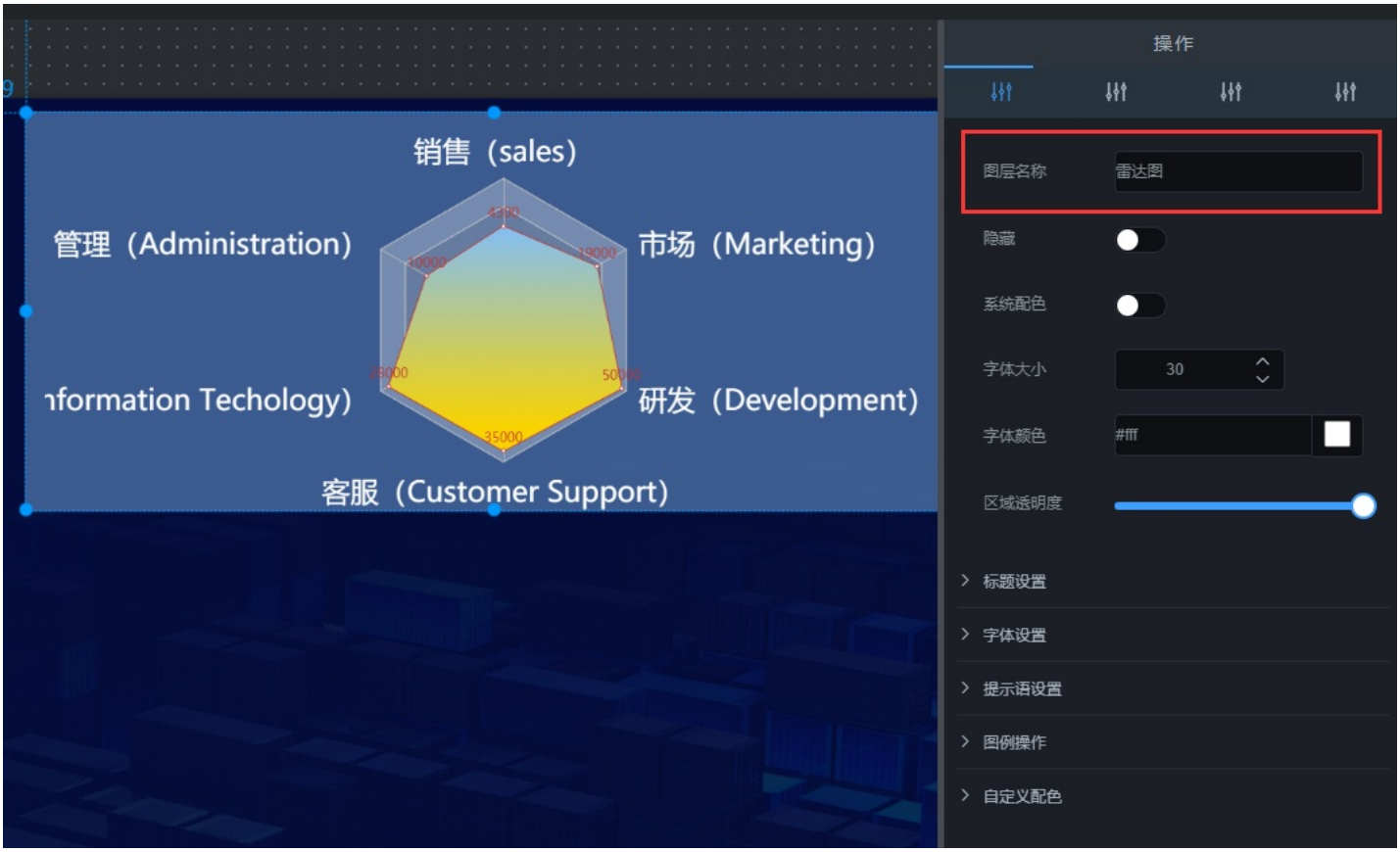


图2. 62

二、系统配色

选中该雷达图组件，在操作界面右侧，打开“系统配色”开关，在“配色选择”下拉框中选择主题，来修改雷达图组件的配色，如图2.63。

** 备注：使用系统配色，需要先删除自定义颜色；**

- 默认配色：效果图如图2.631；
- 紫色主题：效果图如图2.632；
- 绿色主题：效果图如图2.633；

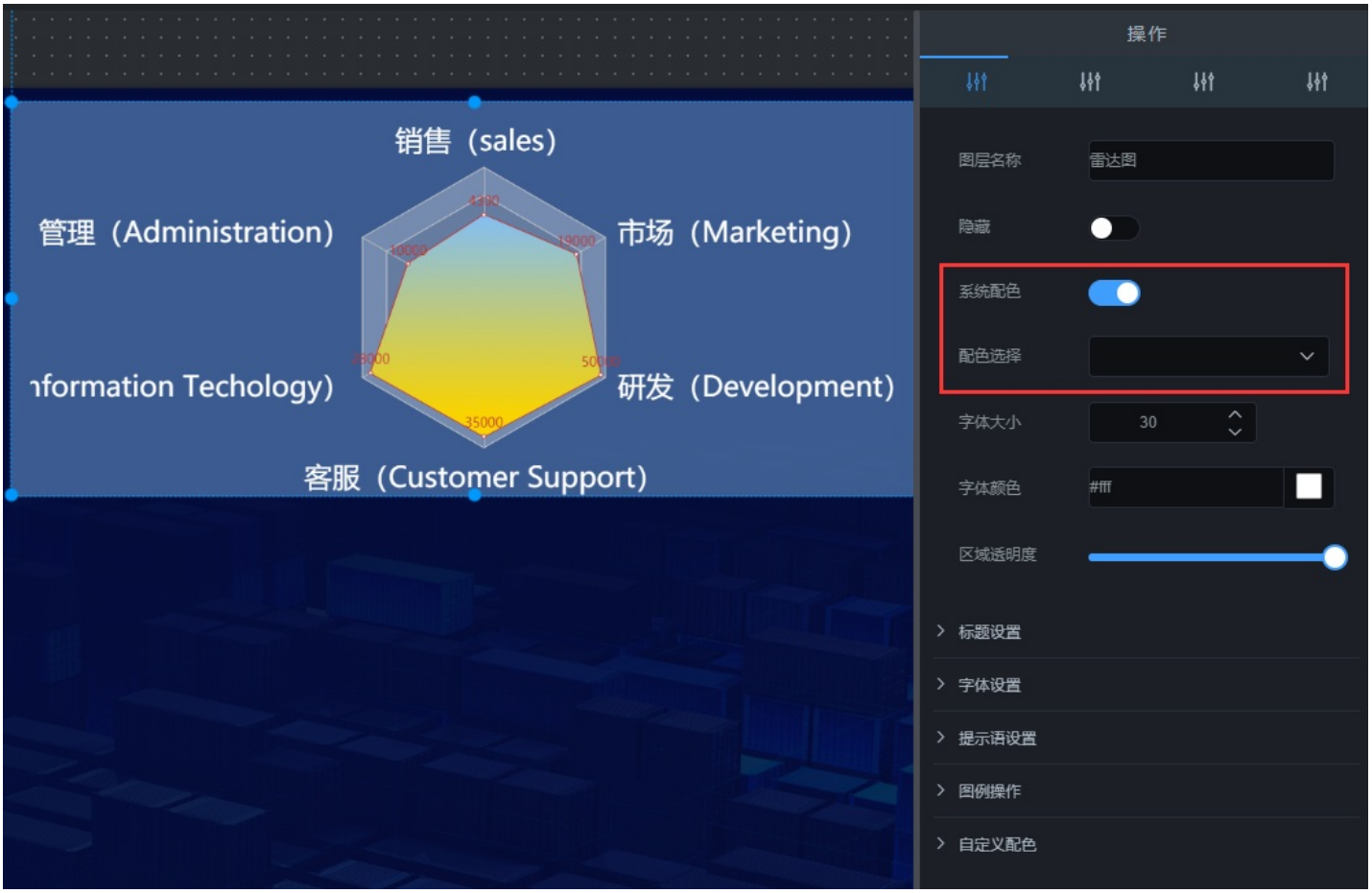


图2. 63

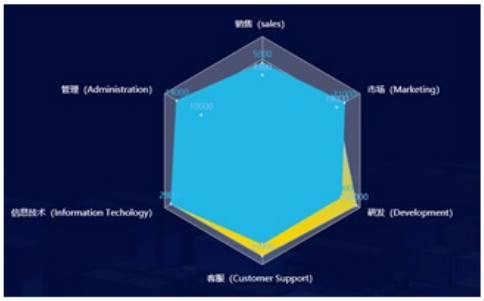


图 2.631

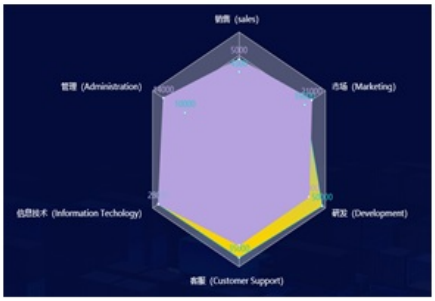


图 2.632

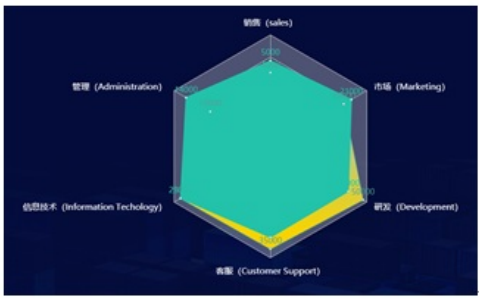


图 2.633

三、字体大小

选中该雷达图，在操作界面右侧的“字体大小”处可修改文字的大小，如图2.64。

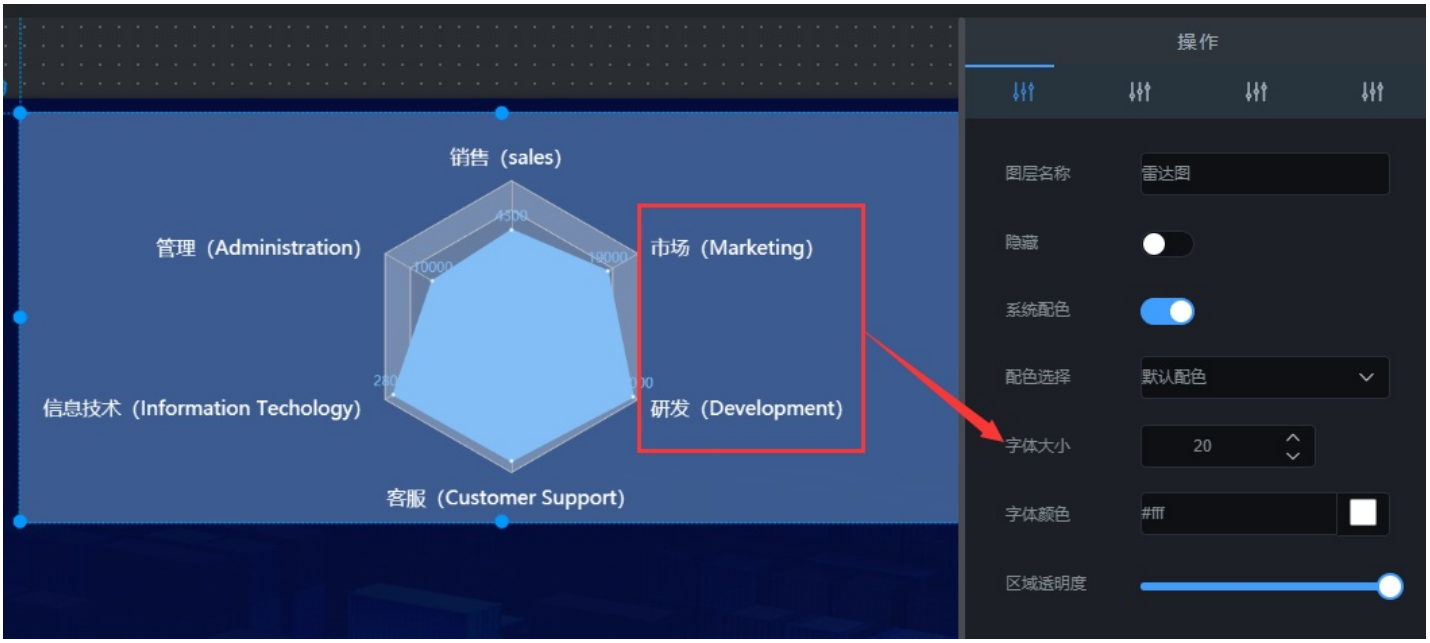


图2.64

四、字体颜色

选中该雷达图，在操作界面右侧的“字体颜色”处可修改文字的颜色，如图2.65。

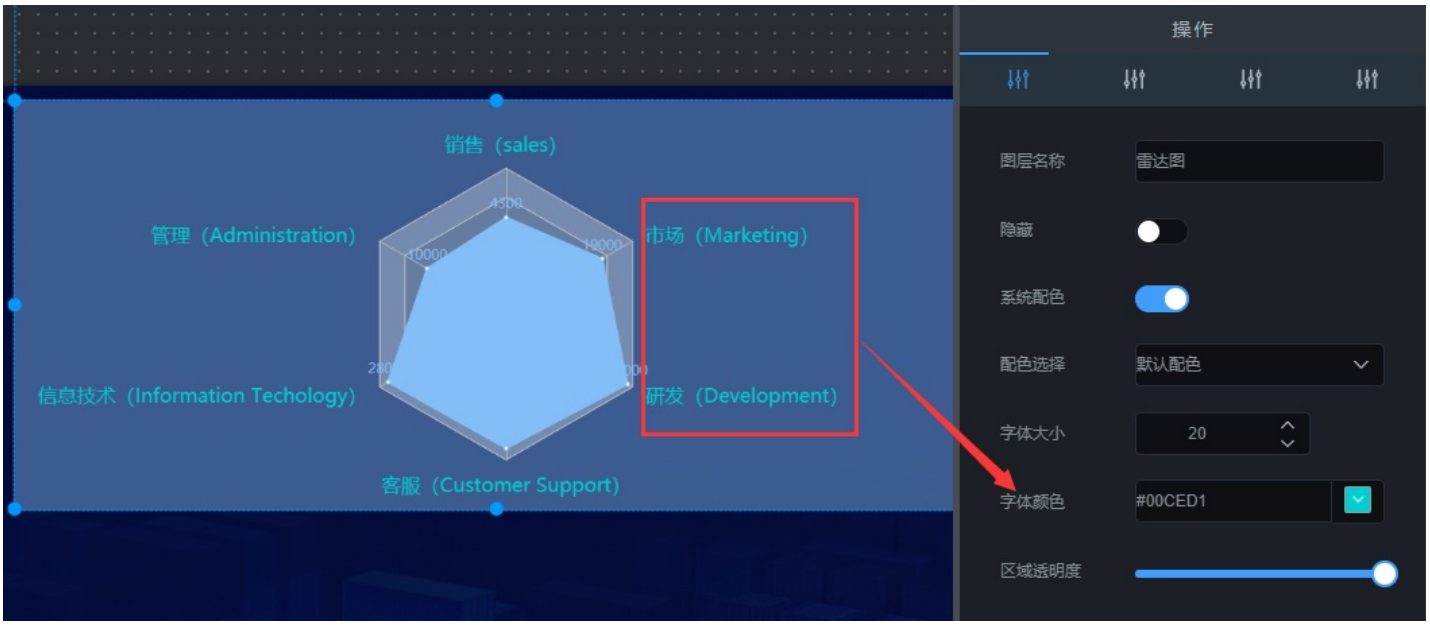


图2.65

五、区域透明度

选中该雷达图组件，在操作界面右侧，拖动“区域透明度”，可设置雷达区域的透明度，如图2.66。

- 0~1，透明度不断变小，0的透明度最大，1透明度最小（即不透明）；

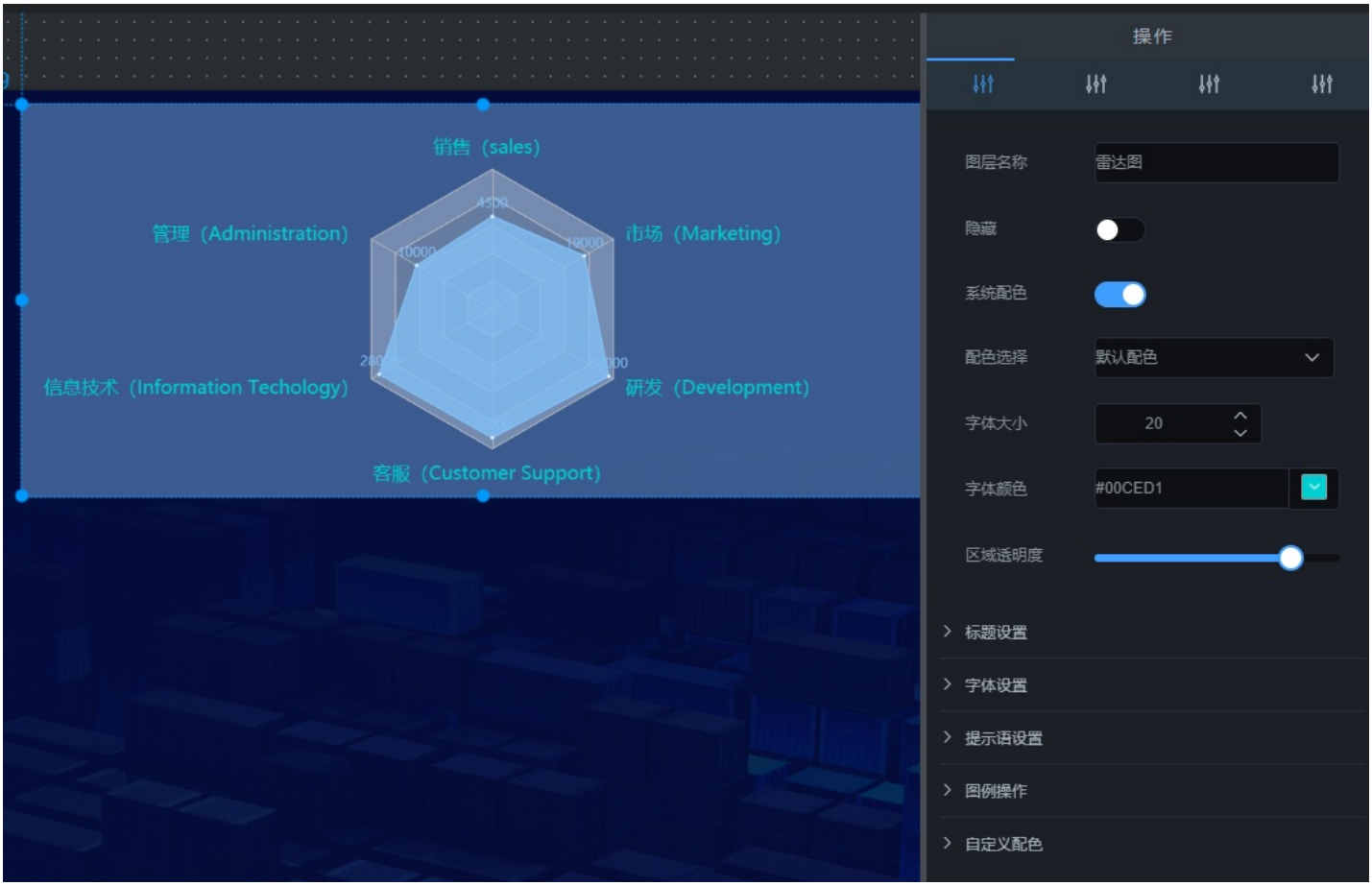


图2.66

六、标题设置

选中该雷达图组件，在操作界面右侧的“标题设置”处可修改雷达图组件的标题样式，如图2.67。

- 标题开关：该开关控制标题的显示与隐藏；
- 标题：标题显示的内容；
- 字体颜色：标题的颜色；
- 字体粗细：标题字体的粗细；
- 字体大小：标题字体大小；
- 字体位置：标题的位置，分为：居中、左对齐、右对齐；
- 副标题：副标题内容；
- 字体颜色：副标题字体颜色；
- 字体粗细：副标题字体的粗细；
- 字体大小：副标题字体大小；

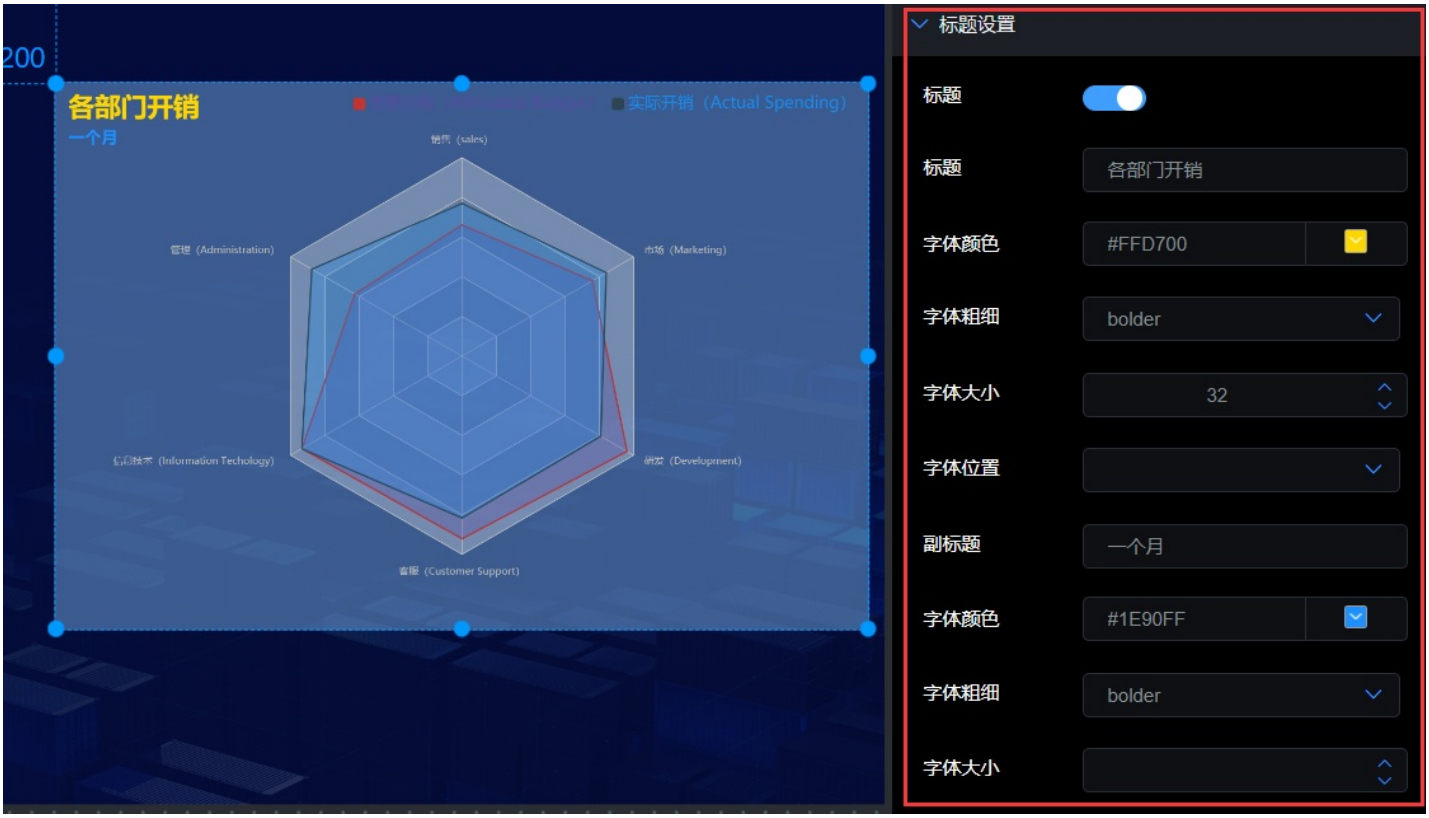


图2.67

七、字体设置

选中该雷达图，在操作界面右侧的“字体设置”处可修改雷达图组件的数字字体样式，如图2.68。

- 显示：数值文字是否显示；
- 字体大小：文字的大小；
- 字体颜色：文字的颜色；
- 字体粗细：文字的粗细；

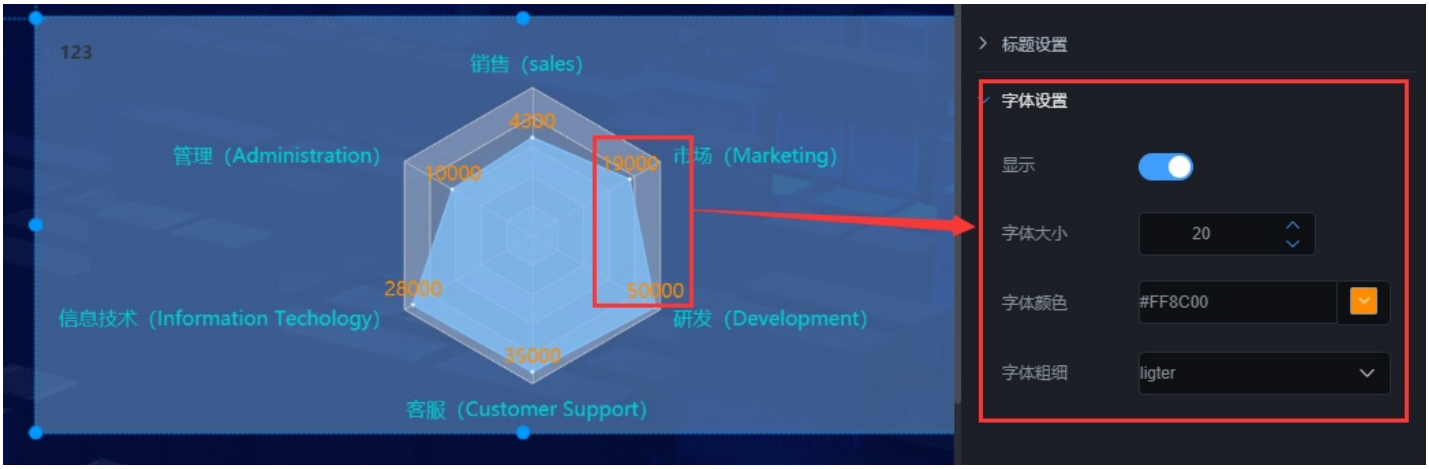


图2.68

八、提示语设置

选中该雷达图组件，在操作界面右侧的“提示语设置”处可修改雷达图组件的提示语，如图2.69。

- 字体大小：提示语的字体大小；
- 字体颜色：提示语的字体颜色；

图2.69

九、图例设置

选中该雷达图组件，在操作界面右侧的“图例设置”处可设置图例的样式，如图2.691；效果图如图2.692。

- 图例开关：是否显示图例；
- 图例高度：图例图标的高度；
- 图例宽度：图例图标的宽度；
- 位置：图例的位置，分为：居中、左对齐、右对齐；
- 布局朝向：图例的排列顺序，分为：横排和竖排；
- 字体大小：图例的字体大小；

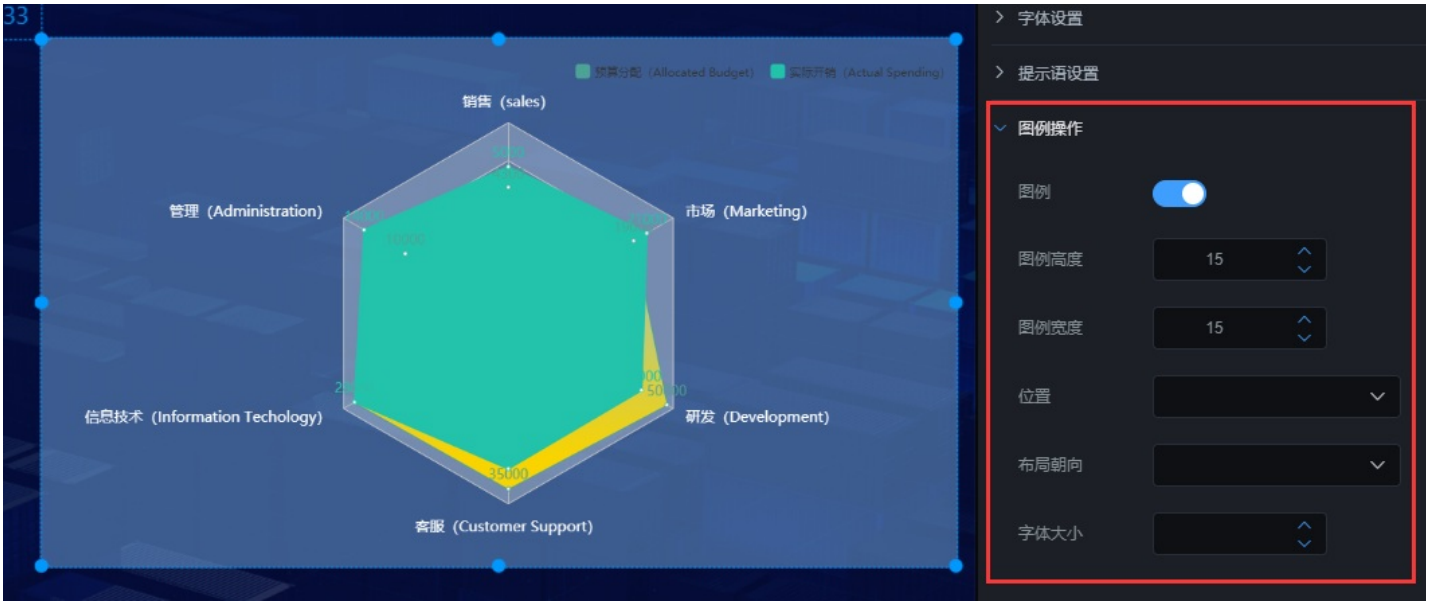


图2.691

图2.692

十、自定义配色设置

选中该折线图组件，在操作界面右侧的“自定义配色设置”处可配置上边不能设置的内容，如图2.692。

- 文字颜色、 轴线颜色暂时不起作用（后期会有版本优化）；
- 配色：雷达覆盖区域的颜色，如果开启了“系统配色” ，需要先把系统配色先关掉，这样自定义的颜色才起作用；

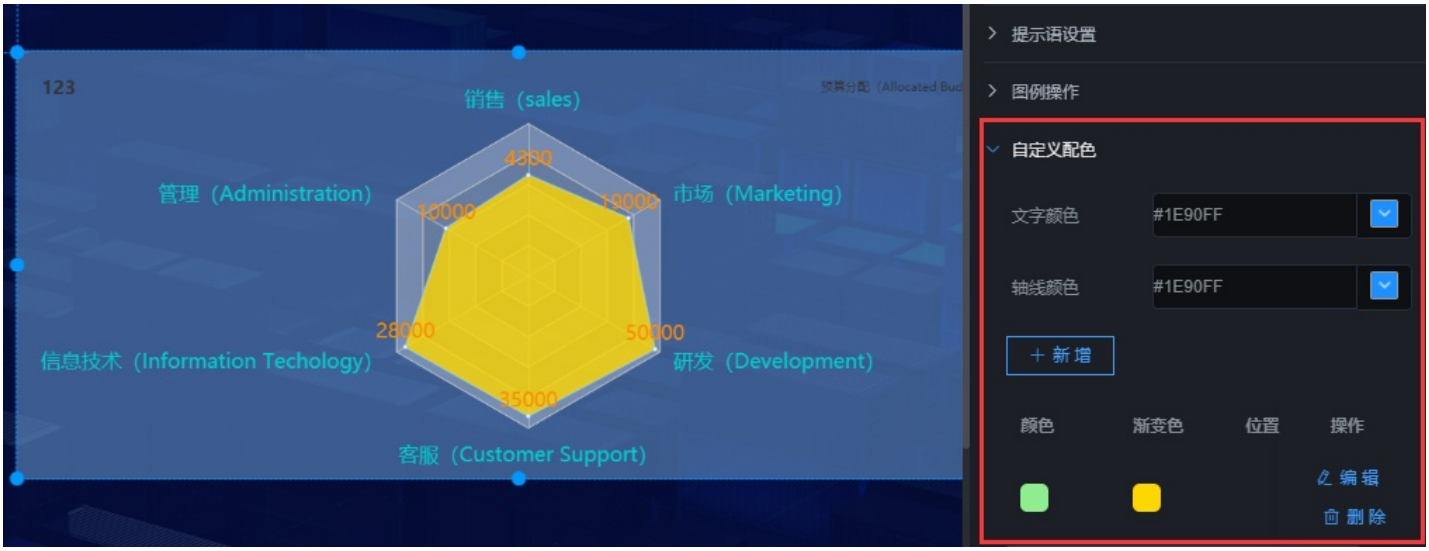


图2.693

十一、接口设置

选中该雷达图组件，在操作界面右侧，点击“

”，可设置接口，如图2.694。

1. 数据类型

数据类型分为静态数据和动态数据；

- 静态数据：写死的数据；
- 动态数据：会随着接口传过来的数据实时变化；（一般这种比较常用）

2. 接口地址

（1）静态数据，接口地址传过来的内容要类似以下格式：

```
{"indicator":[{"name":"销售（sales）","max":6500},{"name":"管理（Administration）","max":16000},{"name":"信息技术（Information Techology）","max":30000},{"name":"客服（Customer Support）","max":38000},{"name":"研发（Development）","max":52000},{"name":"市场（Marketing）","max":25000}],"series":[{"data":[{"value":[4300,10000,28000,35000,50000,19000],"name":"预算分配（Allocated Budget）"}]}]}
```

（2）动态数据，接口地址传过来的内容要类似以下格式：

```
{"data":{"indicator":[{"name":"销售（sales）","max":6500},{"name":"管理（Administration）","max":16000},{"name":"信息技术（Information Techology）","max":30000},{"name":"客服（Customer Support）","max":38000},{"name":"研发（Development）","max":52000},{"name":"市场（Marketing）","max":25000}],"series":[{"data":[{"value":[
```

```
4300,10000,28000,35000,50000,19000],"name":"预算分配（Allocated Budget）"}]}]}}
```

3. 刷新时间

这个参数主要针对动态数据设置的，完成数据的实时更新。

- 如果你想设置成5秒刷新一次，可以将刷新时间设置成“5000”；

4. 刷新数据

这个参数主要是重新请求以下接口，完成数据的更新。


数据类型
静态数据
动态数据
接口地址
/moc
k/26/bar
接口方式
POST
GET
刷新时间
5000
数据处理

图2.964

2.7散点图组件

散点图组件就是添加散点图的组件。点击“ ”图标，再点击“雷达图”，即可创建新的图像，如图2.71；

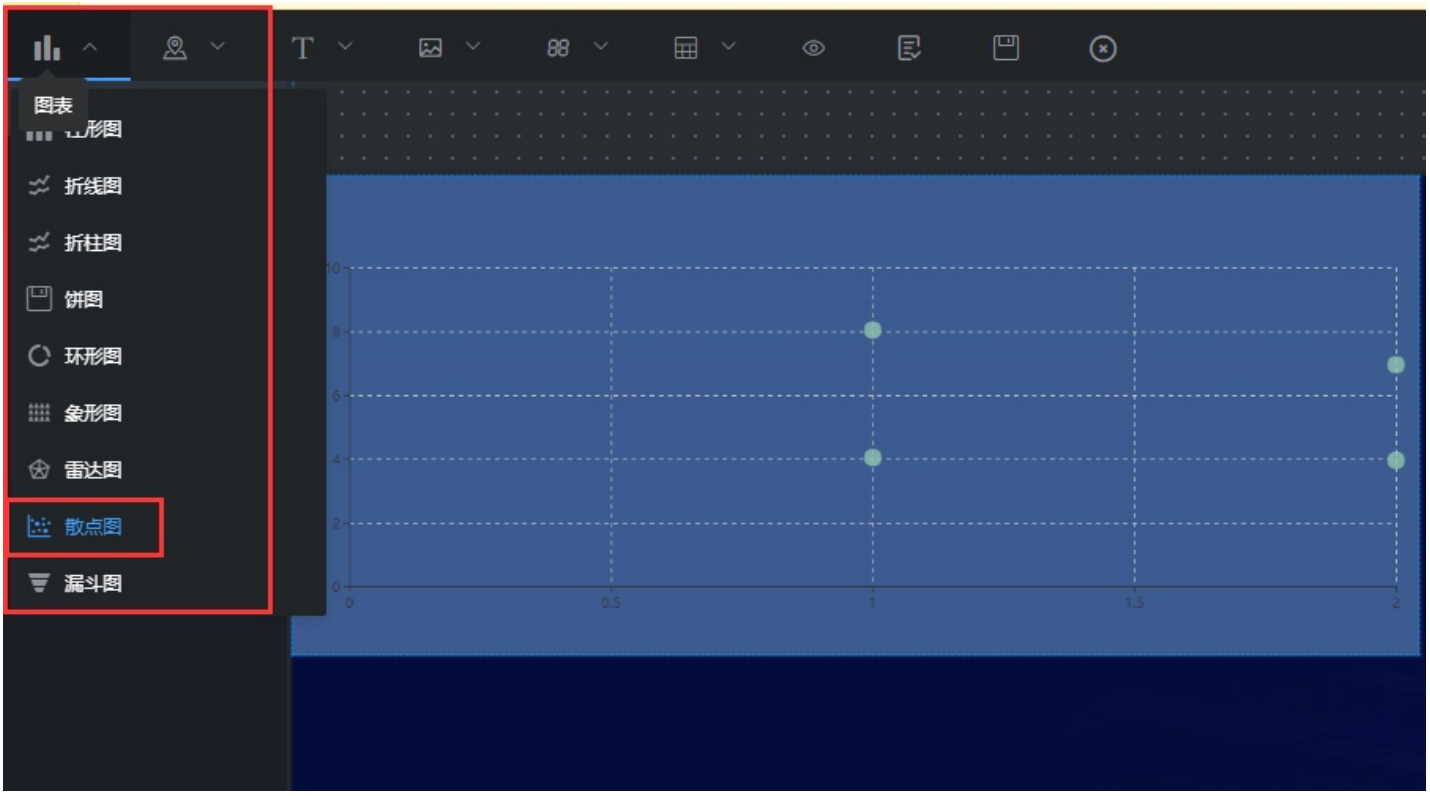


图2.71

一、组件名称设置

选中散点图组件，在操作界面右侧的“图层名称”处可修改文本组件的名称，如图2.72。（名称最好要设置一下，方便后期组件管理）

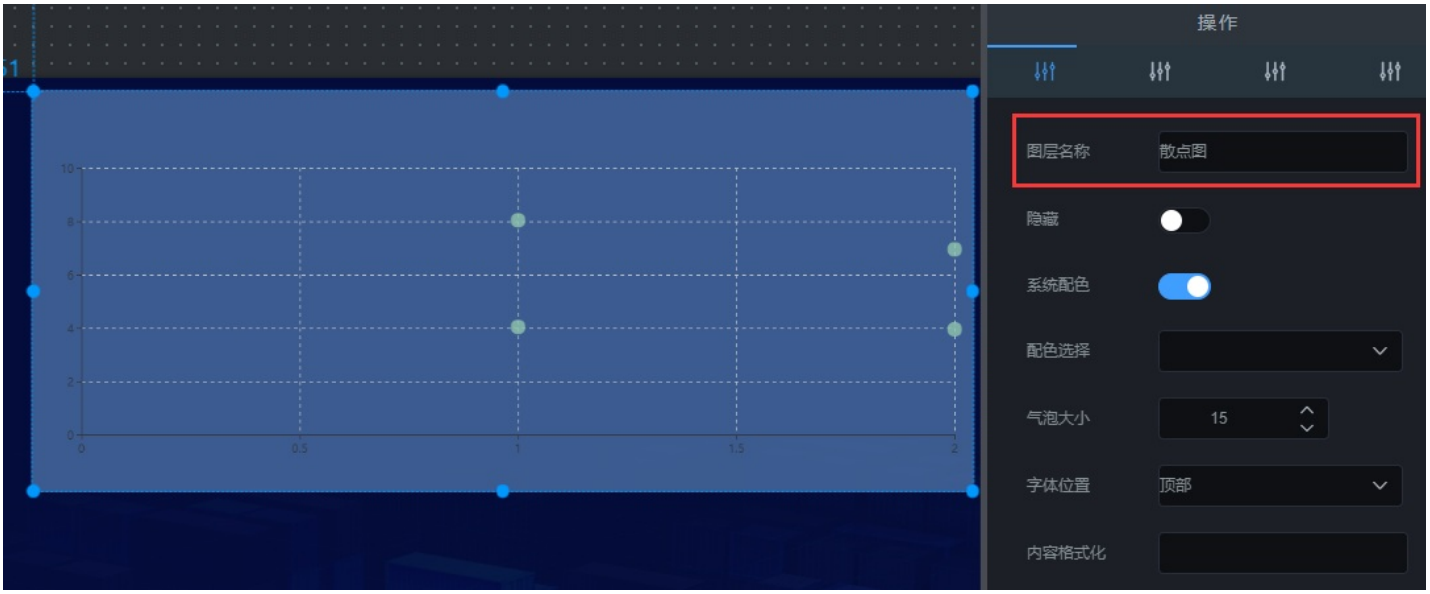


图2. 72

二、系统配色

选中该散点图组件，在操作界面右侧，打开“系统配色”开关，在“配色选择”下拉框中选择颜色，来修改散点图组件的配色，如图2.73。

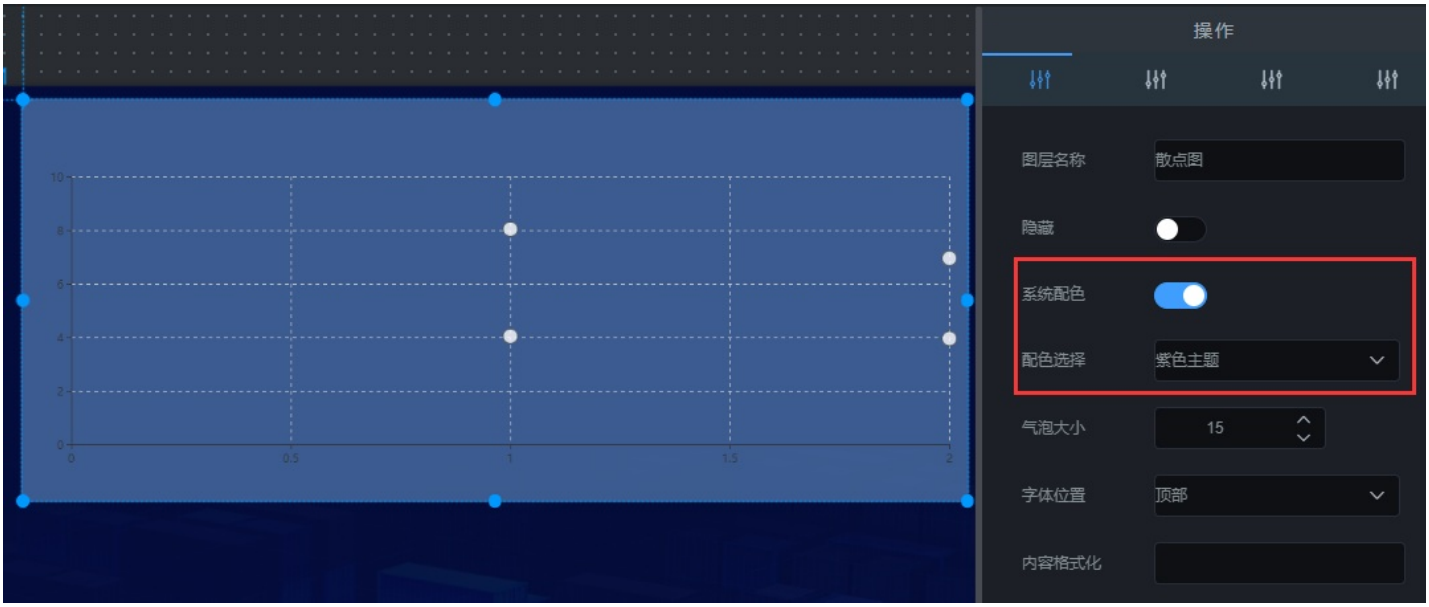


图2. 73

三、气泡大小

选中该散点图，在操作界面右侧的“气泡大小”处可修改点的大小，如图2.74。

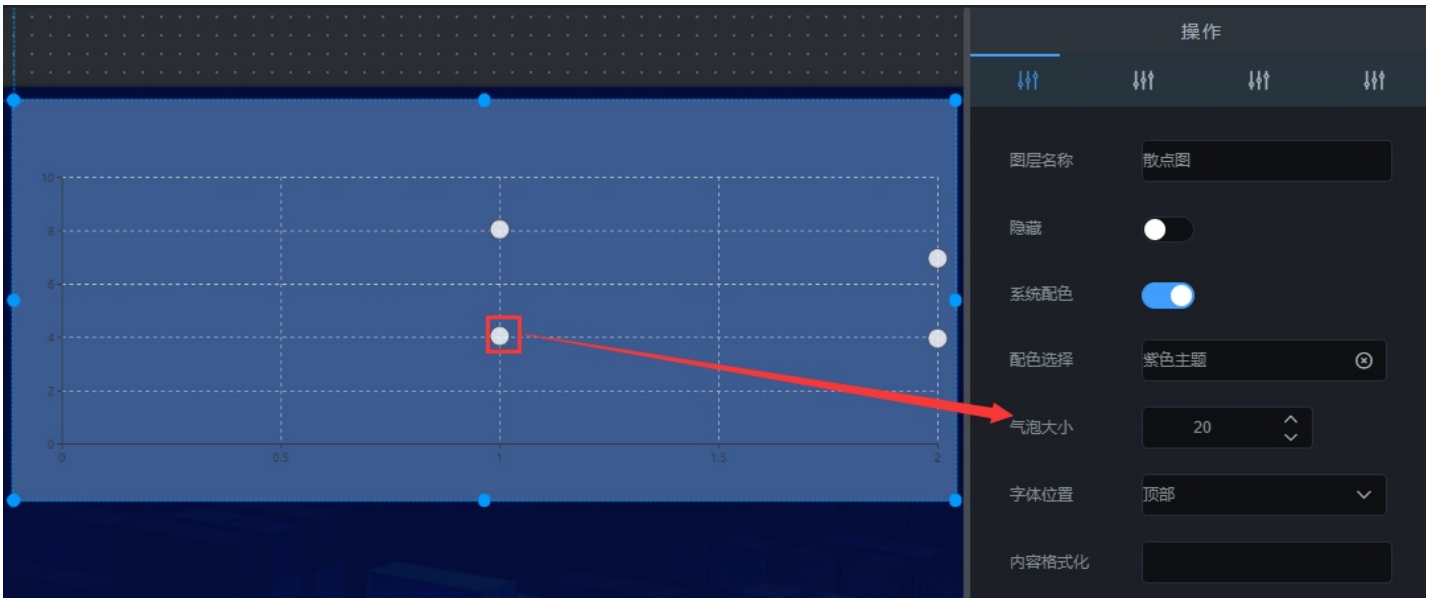


图2. 74

四、字体位置

选中该散点图图，在操作界面右侧，通过设置“字体位置”，来修改字体样式，如图2.75。

- 顶部：字体位于圆点的顶部；
- 内部：字体位于圆点的内部；
- 左部：字体位于圆点的左部；
- 右部：字体位于圆点的右部；

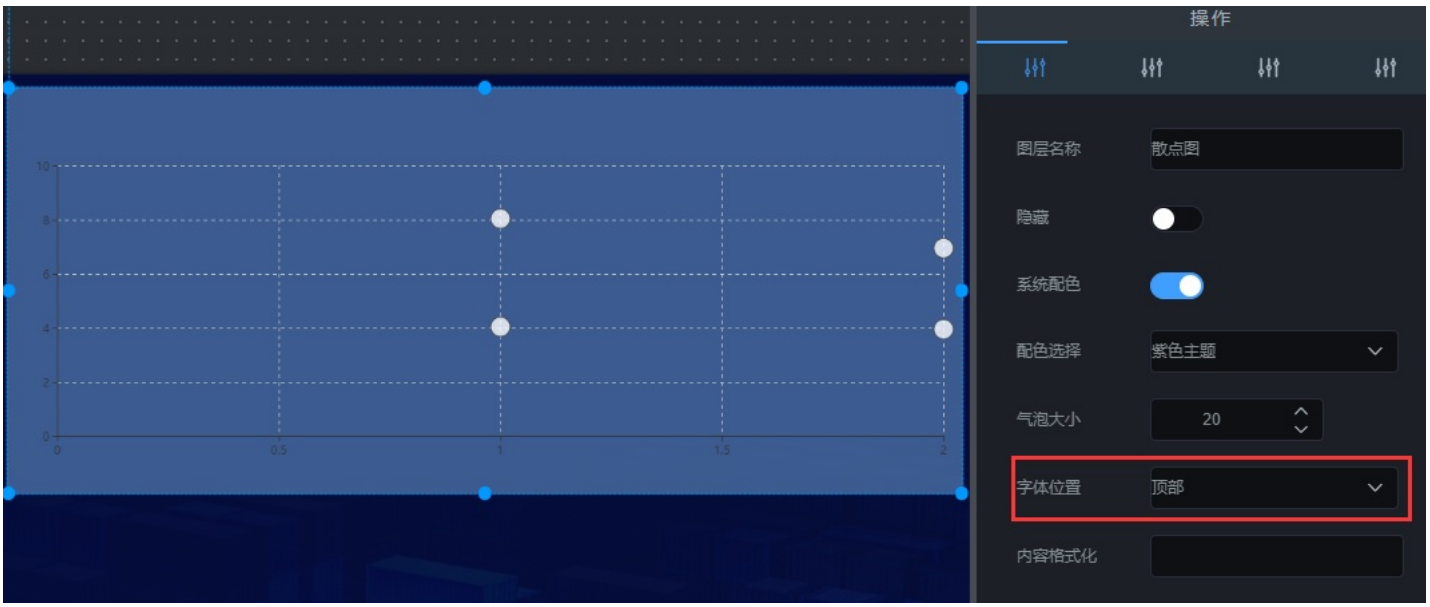


图2.75

五、内容格式化

选中该散点图组件，在操作界面右侧，在内容格式化中输入代码，可格式化样式。

六、X轴设置

选中该散点图组件，在操作界面右侧的“X轴设置”处可修改散点图组件的X轴样式，如图2.76。

- 名称：X轴的名称；
- 显示：X轴是否显示；
- 显示网格：网格是否显示；
- 轴线颜色：网格线颜色设置；
- 标签间距：每个柱状图之间的距离；
- 文字角度：X轴文字的旋转角度；
- 轴反转：柱状图顺序左右调转；
- 字号：X轴字体大小；

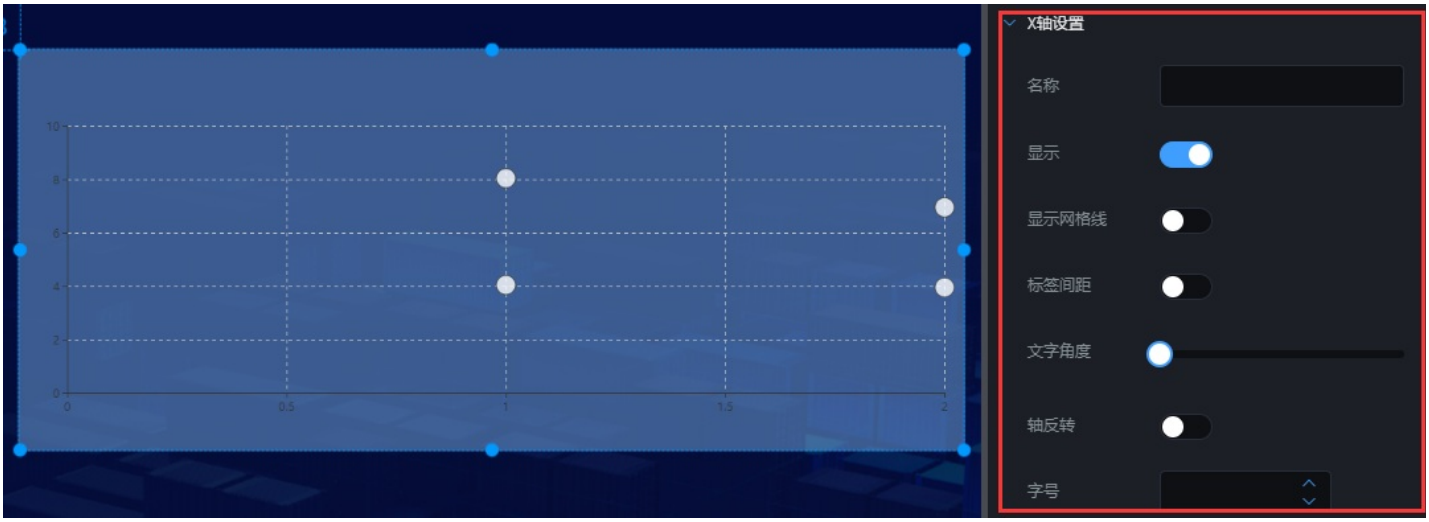


图2.76

七、Y轴设置

选中该散点图组件，在操作界面右侧的“Y轴设置”处可修改散点图组件的Y轴样式，如图2.77。

- 名称：Y轴的名称；
- 显示：Y轴是否显示；
- 轴网格线：网格是否显示；
- 轴线颜色：网格线颜色设置；
- 轴反转：柱状图顺序上下调转；
- 字号：Y轴字体大小；

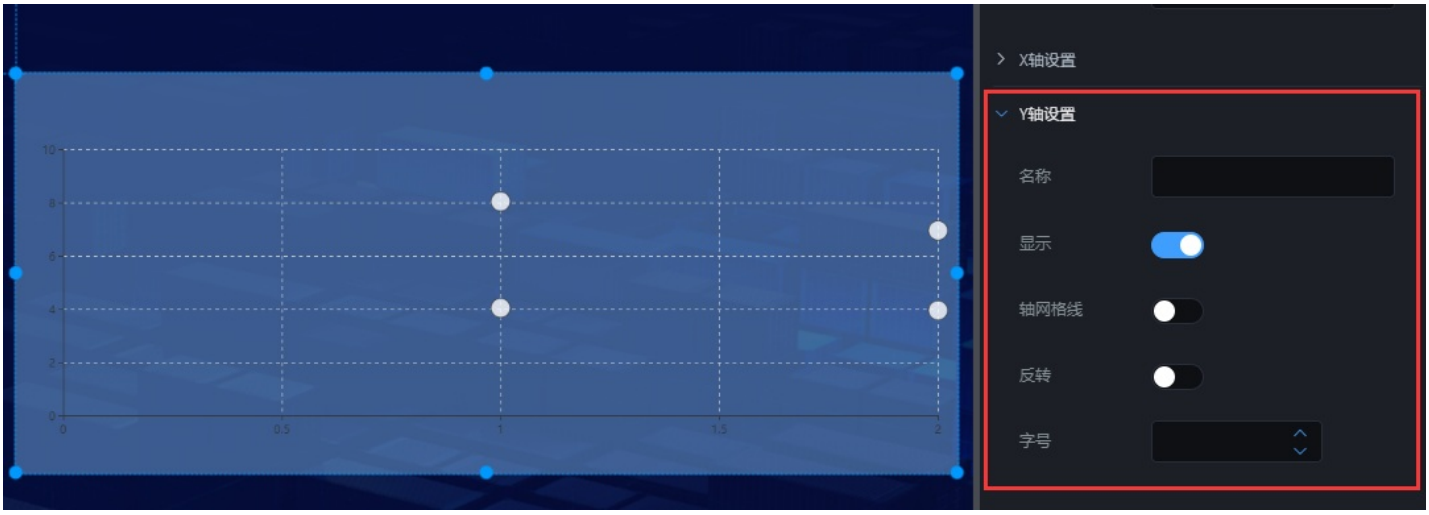


图2.77

八、字体设置

选中该散点图，在操作界面右侧的“字体设置”处可修改散点图组件的字体样式，如图2.78。

- 显示：数值文字是否显示；
- 字体大小：文字的大小；
- 字体颜色：文字的颜色；
- 字体粗细：文字的粗细；

图2.78

九、提示语设置

选中该散点图组件，在操作界面右侧的“提示语设置”处可修改散点图组件的提示语，如图2.79。

- 字体大小：提示语的字体大小；
- 字体颜色：提示语的字体颜色；

图2.79

十、自定义配色设置

选中该散点图组件，在操作界面右侧的“自定义配色”处可配置上边不能设置的内容，如图2.791。

- 文字颜色、 轴线颜色暂时不起作用（后期会有版本优化）；
- 配色：雷达覆盖区域的颜色；

图2.791

十一、接口设置

选中该雷达图组件，在操作界面右侧，点击“

”，可设置接口，如图2.792。

1. 数据类型

数据类型分为静态数据和动态数据；

- 静态数据：写死的数据；
- 动态数据：会随着接口传过来的数据实时变化；（一般这种比较常用）

2. 接口地址

###（1）静态数据，接口地址传过来的内容要类似以下格式：

```
{"data":[{"data":[[1,8.04],[2,6.95]]},{"data":[[1,4.04],[2,3.95]]}]}
```

###（2）动态数据，接口地址传过来的内容要类似以下格式：

```
{"data":[{"data":[[1,8.04],[2,6.95]]},{"data":[[1,4.04],[2,3.95]]}]}
```

3. 刷新时间

这个参数主要针对动态数据设置的，完成数据的实时更新。

- 如果你想设置成5秒刷新一次，可以将刷新时间设置成“5000”；

4. 刷新数据

这个参数主要是重新请求以下接口，完成数据的更新。


数据类型
静态数据
动态数据
接口地址
[redacted]/moc
k/26/bar
接口方式
POST
GET
刷新时间
5000
数据处理

图2. 792

2.8漏斗图组件

漏斗图组件就是添加漏斗图样式的组件。点击“ ”图标，再点击“漏斗图”，即可创建新的图像，如图2.81；

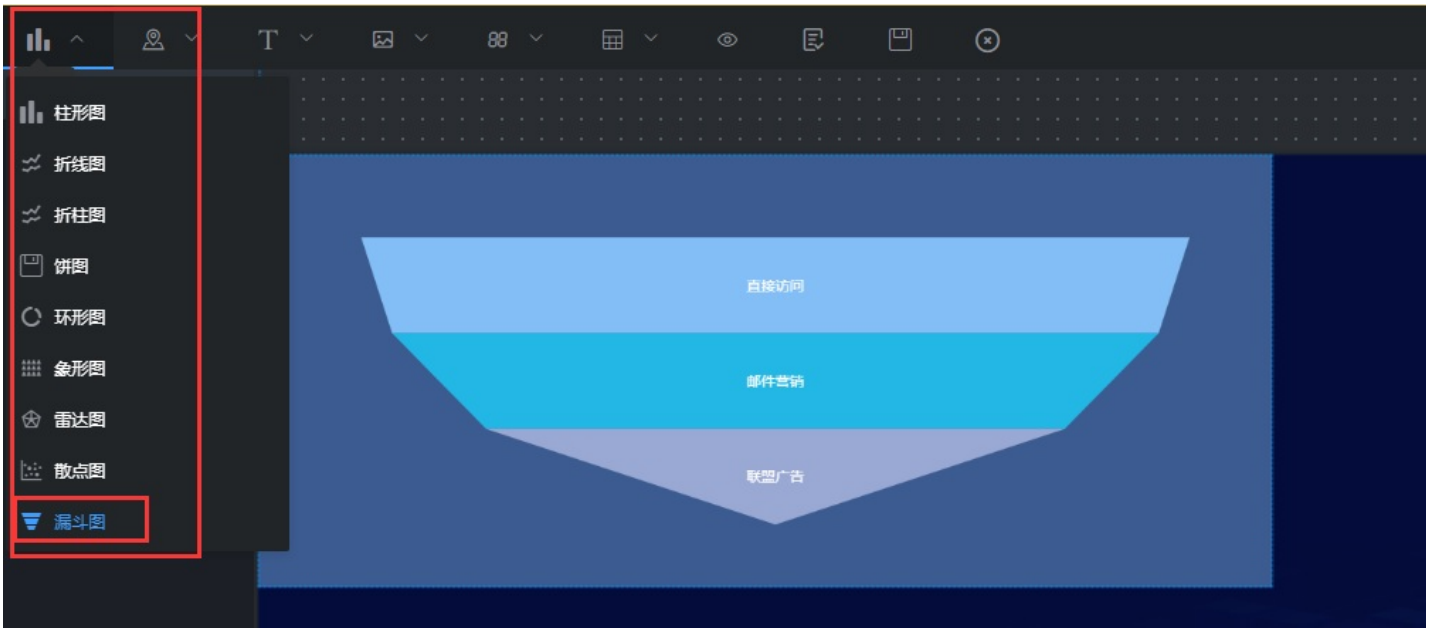


图2.81

一、组件名称设置

选中漏斗图组件，在操作界面右侧的“图层名称”处可修改组件的名称，如图2.82。（名称最好要设置一下，方便后期组件管理）

图2.82

二、系统配色

选中该漏斗图组件，在操作界面右侧，打开“系统配色”开关，在“配色选择”下拉框中选择主题，来修改漏斗图组件的配色，如图2.83。

图2. 83

三、文字设置

选中该漏斗图组件，在操作界面右侧的“文字设置”处可修改漏斗图组件内文字样式，如图2.84。

- 显示开关：该开关控制内部文字的显示与隐藏；
- 字体大小：设置内部文字的大小；
- 字体颜色：设置内部文字的颜色；
- 字体粗细：设置内部文字的粗细；

图2.84

四、反转

选中该漏斗图组件，在操作界面右侧，打开“反转”开关，可将漏斗图反转180度，如图2.85。（效果图2.851）

图2.85

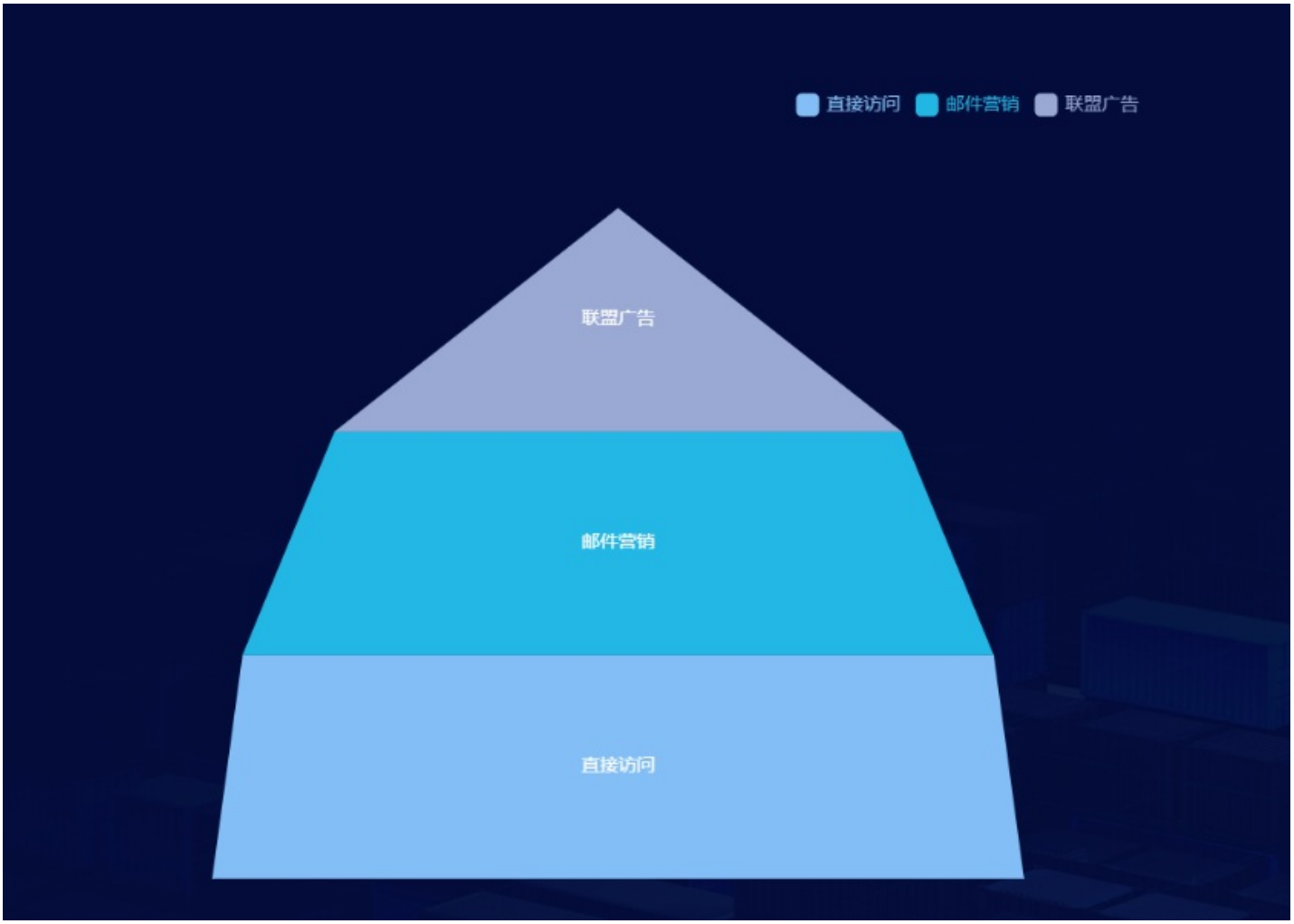


图2.851

五、标题设置

选中该漏斗图组件，在操作界面右侧的“标题设置”处可修改漏斗图组件的标题样式，如图2.86。

- 标题开关：该开关控制标题的显示与隐藏；
- 标题：标题显示的内容；
- 字体颜色：标题的颜色；
- 字体粗细：标题字体的粗细；
- 字体大小：标题字体大小；
- 字体位置：标题的位置，分为：居中、左对齐、右对齐；
- 副标题：副标题内容；
- 字体颜色：副标题字体颜色；
- 字体粗细：副标题字体的粗细；
- 字体大小：副标题字体大小；

图2. 86

六、字体设置

选中该漏斗图，在操作界面右侧的“字体设置”处可修改漏斗图组件的文字样式，如图2.87。

- 显示：文字是否显示；
- 字体大小：文字的大小；
- 字体颜色：文字的颜色；
- 字体粗细：文字的粗细；

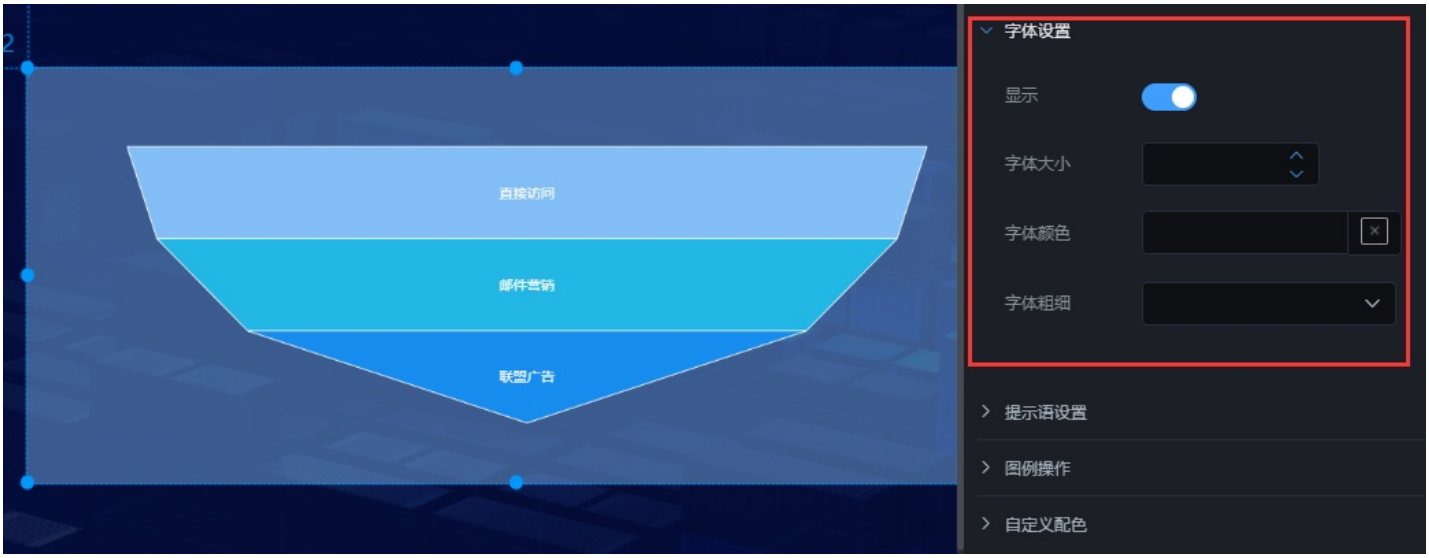


图2.87

七、提示语设置

选中该漏斗图组件，在操作界面右侧的“提示语设置”处可修改漏斗图组件的提示语样式，如图2.88。

- 字体大小：提示语的字体大小；
- 字体颜色：提示语的字体颜色；

图2.88

八、图例设置

选中该漏斗图组件，在操作界面右侧的“图例设置”处可设置图例的样式，如图2.89。

- 图例开关：是否显示图例；
- 位置：图例的位置，分为：居中、左对齐、右对齐；
- 布局朝向：图例的排列顺序，分为：横排和竖排；

- 字体大小：图例的字体大小；

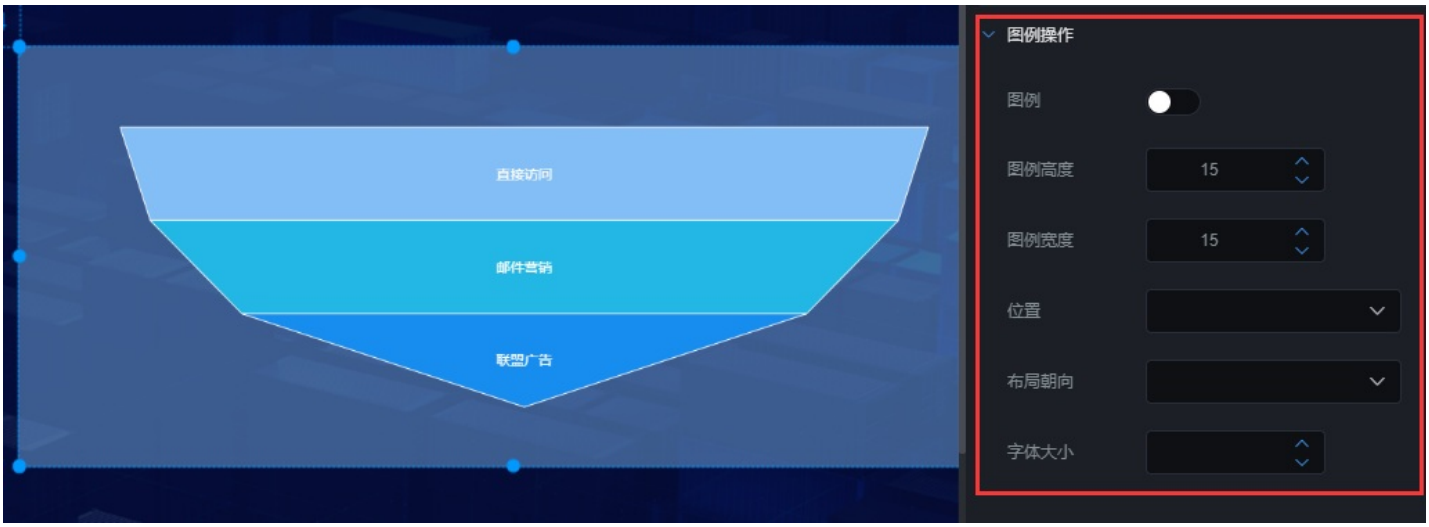


图2.89

九、自定义配色设置

选中该漏斗图组件，在操作界面右侧的“自定义配色”处可配置上边不能设置的内容，如图2.891。

- 文字颜色：轴文字的颜色；（因为漏斗图没有轴，所以不需要设置）
- 轴线颜色：轴线颜色；（因为漏斗图没有轴，所以不需要设置）
- 配色：漏斗图每层的颜色，如果开启了“系统配色”，需要先把系统配色先关掉，这样自定义的颜色才起作用；

图2.891

八、接口设置

选中该漏斗图组件，在操作界面右侧，点击“  ”，可设置接口，如图2.892。

1. 数据类型

数据类型分为静态数据和动态数据；

- 静态数据：写死的数据；
- 动态数据：会随着接口传过来的数据实时变化；

2. 接口地址

（1）静态数据，接口地址传过来的内容要类似以下格式：

```
[{"value":335,"name":"直接访问"},{"value":310,"name":"邮件营销"},{"value":234,"na
me":"联盟广告"}]
```

（2）动态数据，接口地址传过来的内容要类似以下格式：

```
{"data":[{"value":335,"name":"直接访问"},{"value":310,"name":"邮件营销"},{"value"
:234,"name":"联盟广告"}]}
```

3. 刷新时间

这个参数主要针对动态数据设置的，完成数据的实时更新。

- 如果你想设置成5秒刷新一次，可以将刷新时间设置成“5000”；

4. 刷新数据

这个参数主要是重新请求以下接口，完成数据的更新。


数据类型
静态数据
动态数据
接口地址
moc
k/26/bar
接口方式
POST
GET
刷新时间
5000
数据处理

图2.892

3.文本类组件

3.1文本框组件

3.2跑马灯组件

3.3超链接组件

3.4实时时间组件

3.1文本框组件

文本组件就是添加文本的组件。点击“T”图标，再点击“文本框”，即可完成新文本，如图3.11；

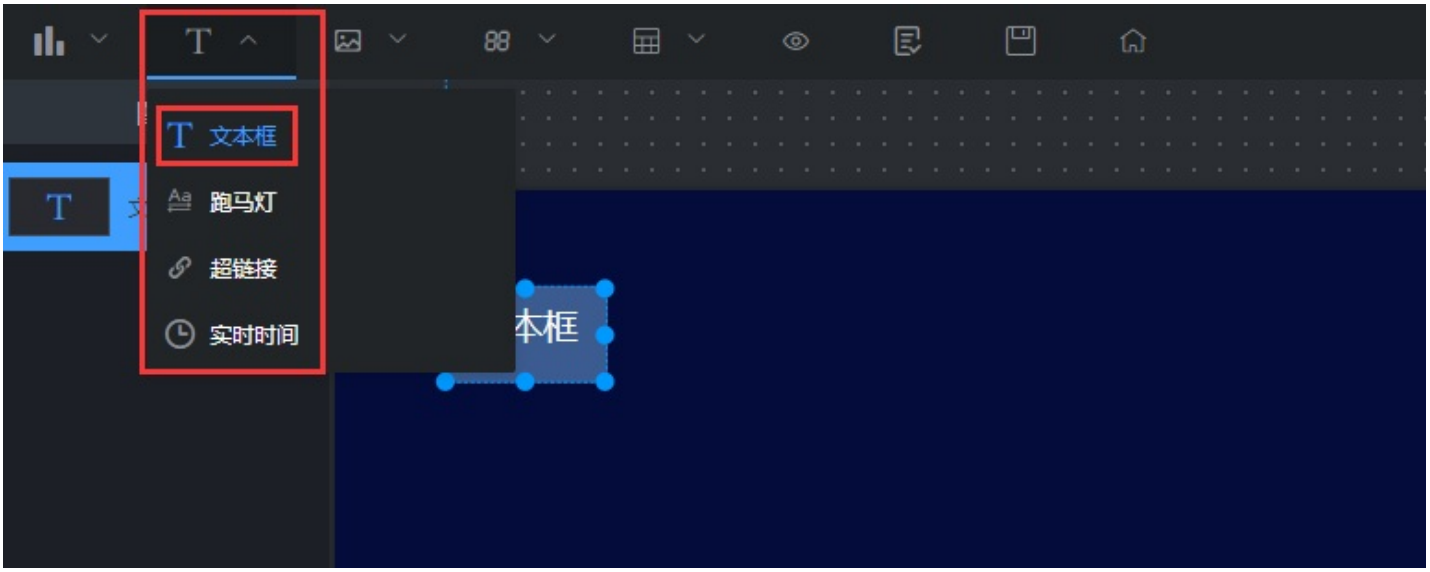


图3. 11

一、组件名称设置

选中该文本组件，在操作界面右侧的“图层名称”处可修改组件的名称，如图3.12。（名称最好要设置一下，方便后期组件管理）

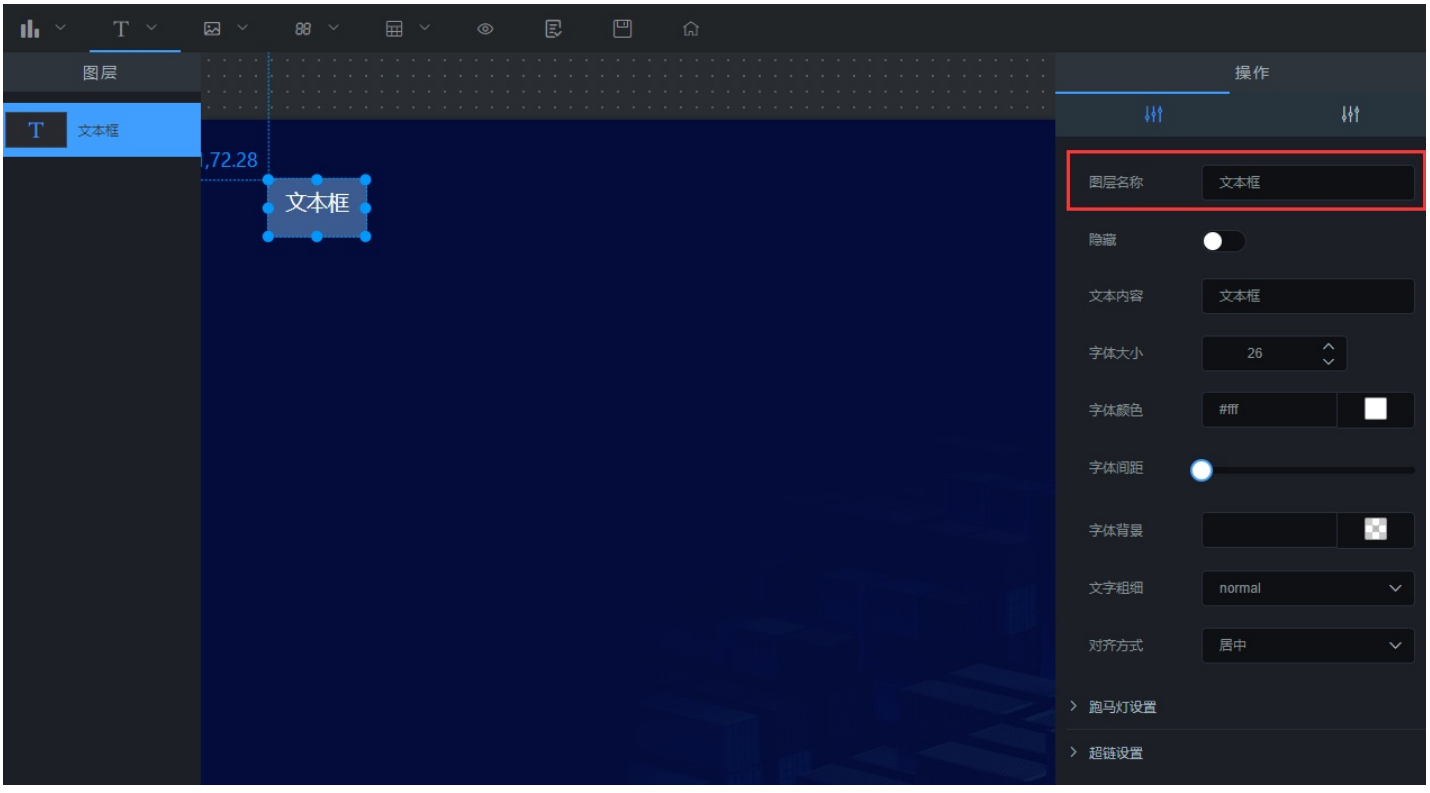


图3. 12

二、内容设置

选中该文本组件，在操作界面右侧的“文本内容”处可修改文本组件的内容，如图3.13。

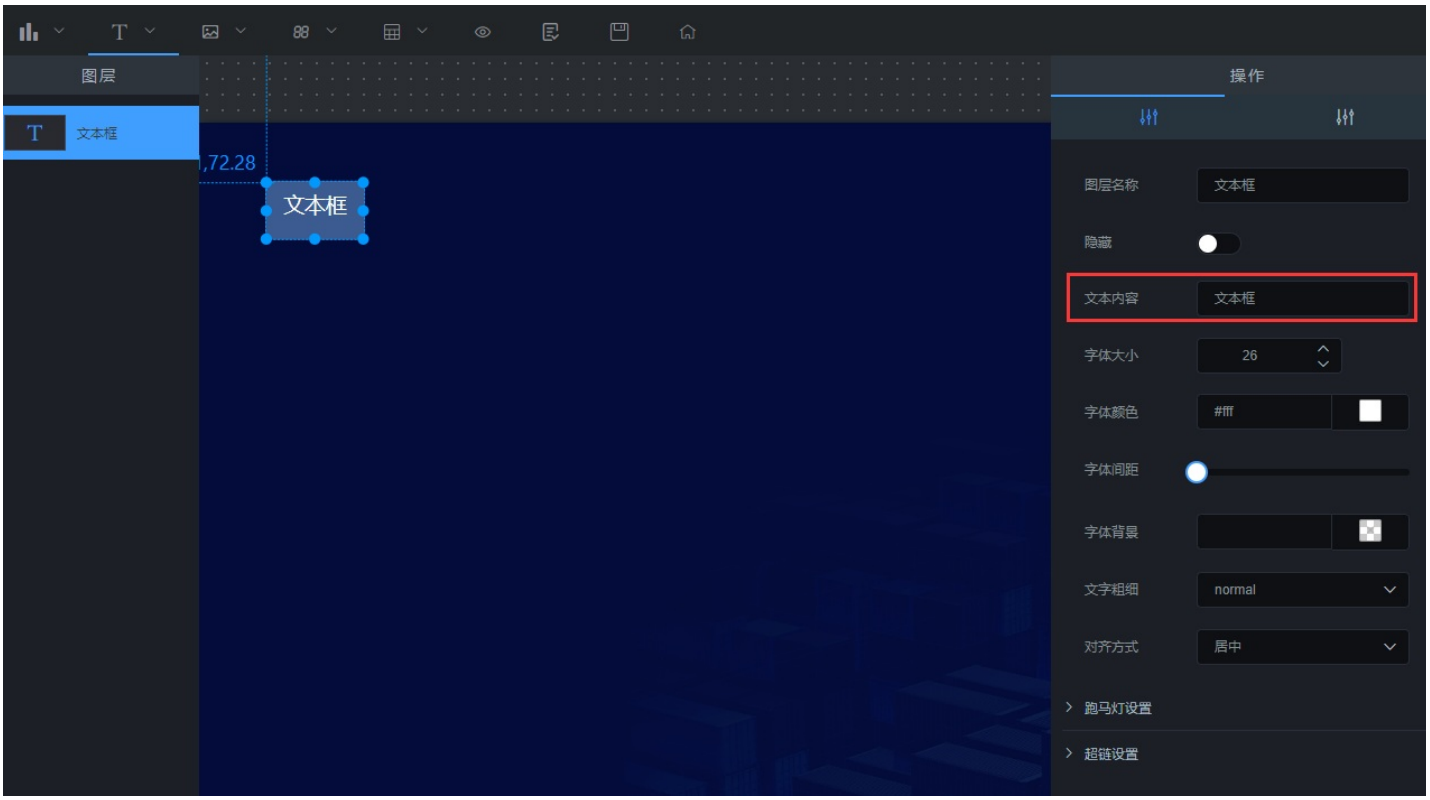


图3. 13

三、 字体样式

- 字体大小：文字字体大小设置；
- 字体颜色：文字颜色设置；
- 字体间距：字之间的间距设置；
- 字体背景：文字背景设置；
- 文字粗细：文字的粗细设置；

图3. 14

四、跑马灯特效

选中该文本组件，在操作界面右侧的“跑马灯设置”处可设置文本的跑马灯特效，如图3.15；

- 首先，点击“开启”按钮，然后设置“滑动速度”即可；

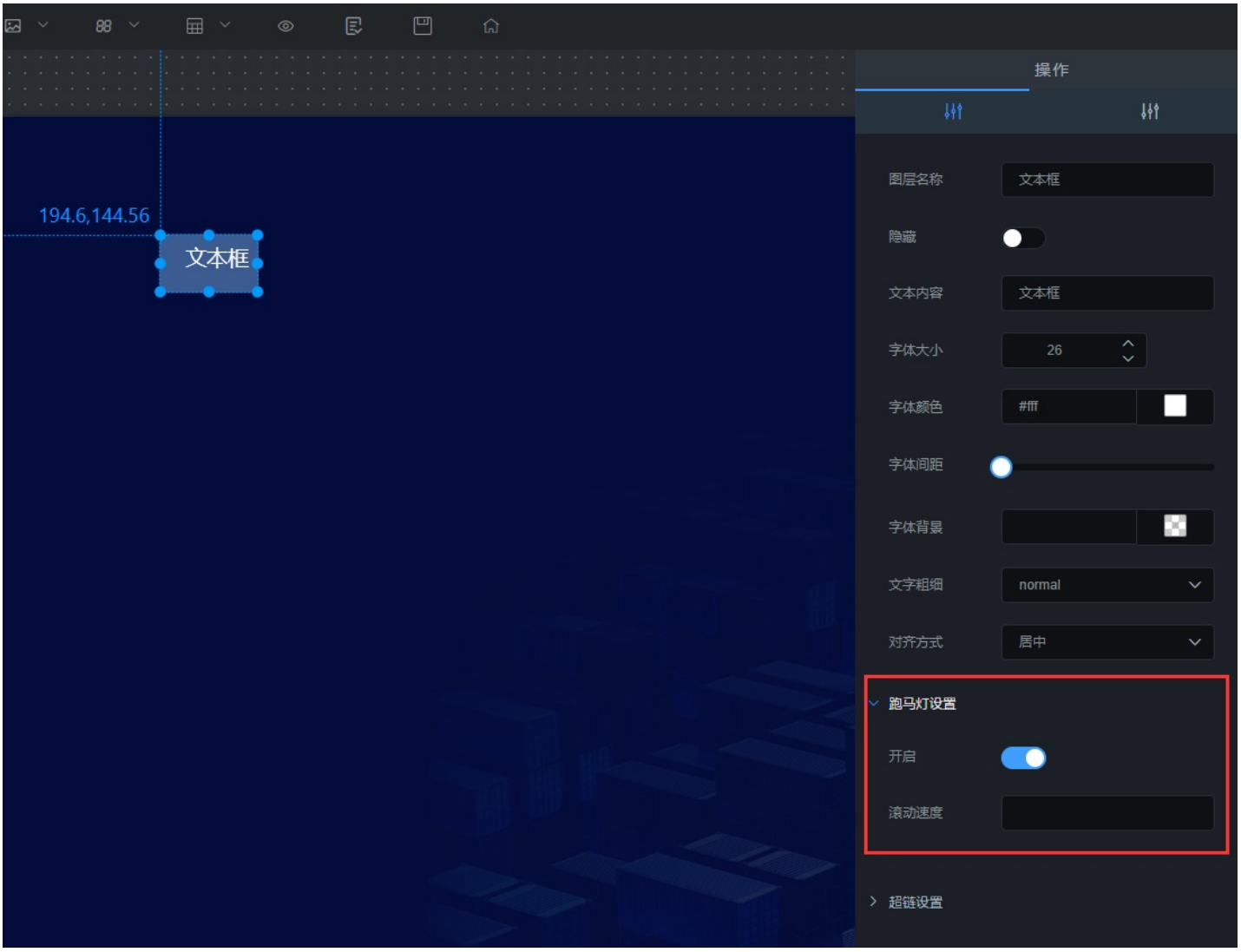


图3.15

五、超链接特效

选中该文本组件，在操作界面右侧的“超链接设置”处可设置文本的跑马灯特效，如图3.16；

- 首先，点击“开启”按钮，然后设置“打开方式”，输入超链地址就可以了；
- 超链地址必须带“http://”或“https://”,如：“http://www.baidu.com”；

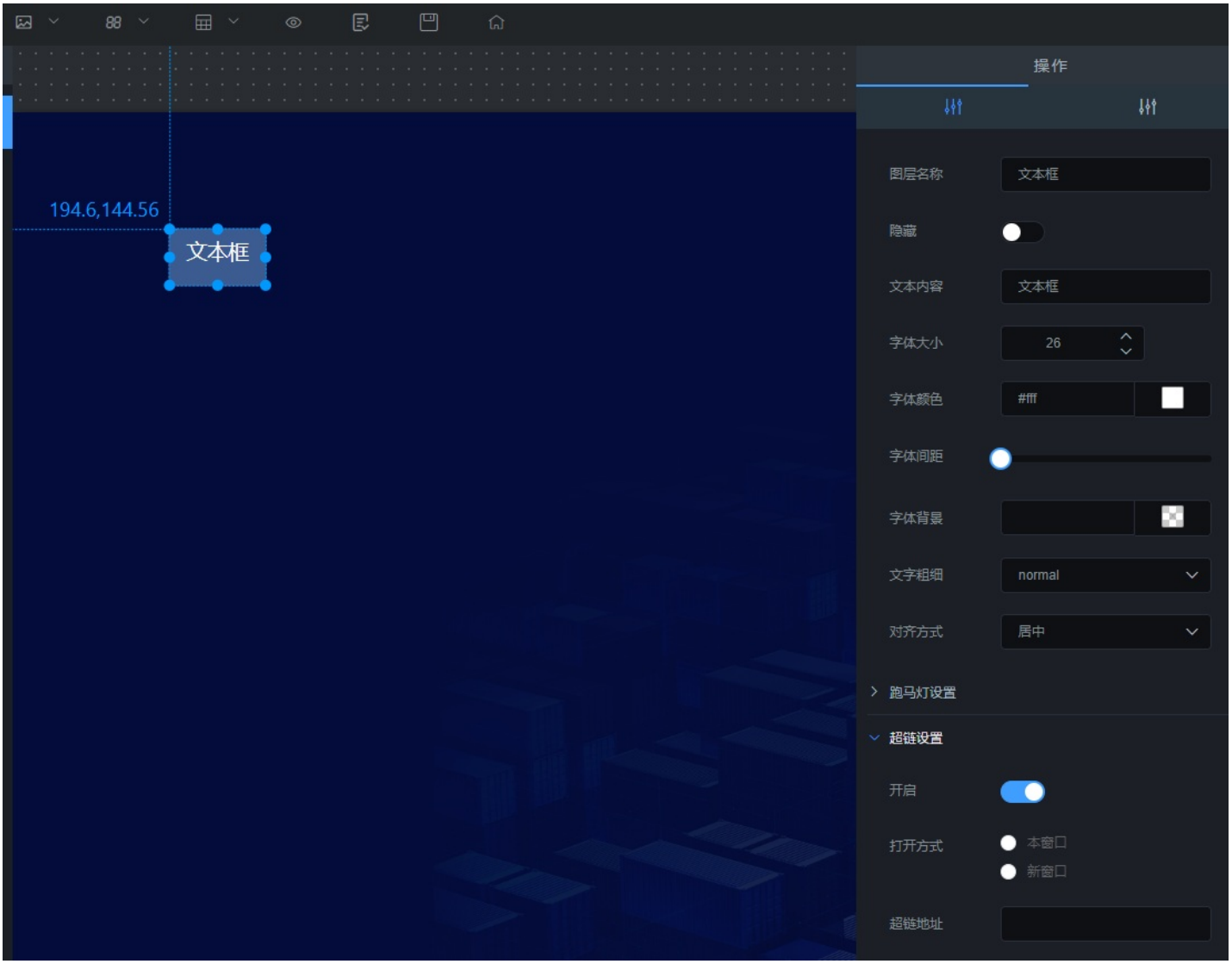

194.6,144.56
文本框
操作
图层名称
文本框
隐藏
文本内容
文本框
字体大小
26
字体颜色
#fff
字体间距
字体背景
文字粗细
normal
对齐方式
居中
跑马灯设置
超链设置
开启
打开方式
本窗口
新窗口
超链地址

图3. 16

3.2跑马灯组件

具体设置跟文本框组件基本一样，可参考文本框组件的设置。

3.3超链接组件

具体设置跟文本框组件基本一样，可参考文本框组件的设置。

3.4实时时间组件

实时时间组件就是设置时间的组件。点击“T”图标，再点击“实时时间”，即可创建时间，如图3.41；

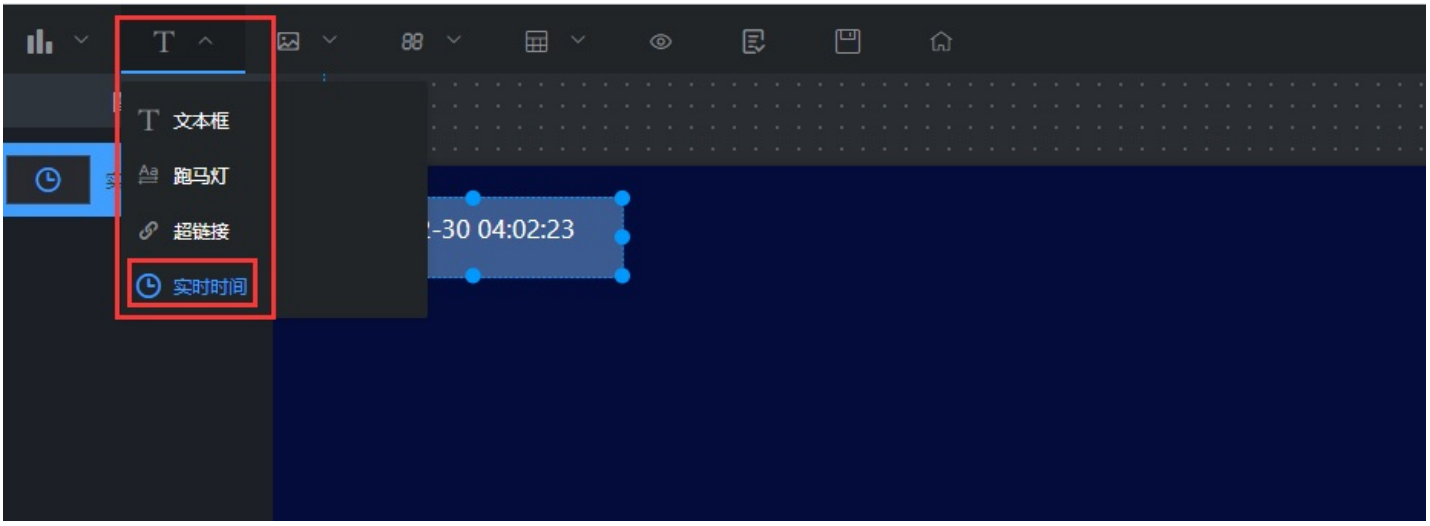


图3.41

一、组件名称设置

选中该实时时间组件，在操作界面右侧的“图层名称”处可修改组件的名称，如图3.42。（名称最好要设置一下，方便后期组件管理）

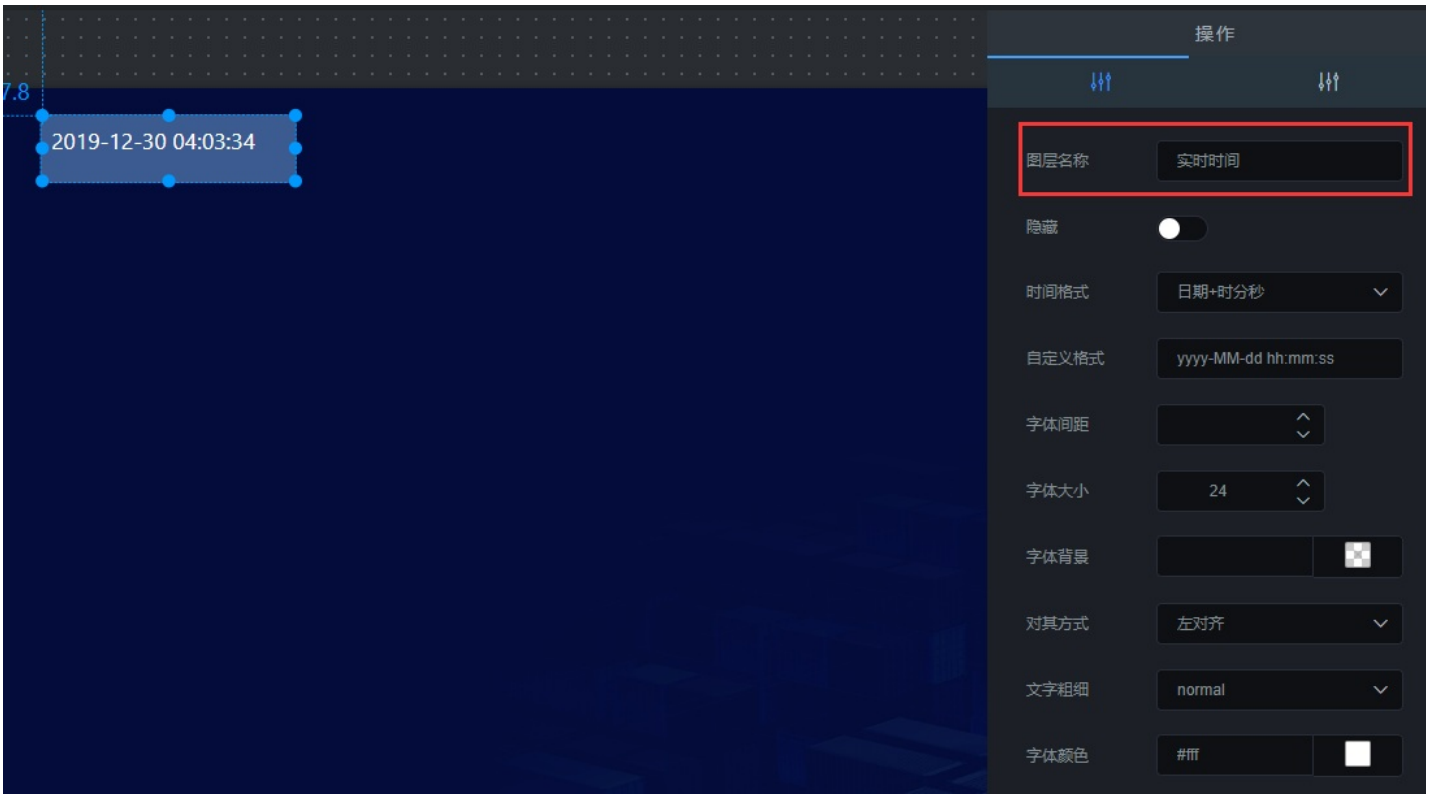


图3.42

二、格式设置

选中该实时时间组件，在操作界面右侧选择“时间格式”，再在“自定义格式”处修组件的格式，如3.43。

- 如果你想设置时间为：2020年01月11日，可选择时间格式为“日期”，再将“自定义格式”处样式设置成“yyyy年MM月dd日”，效果图如3.431；
- 如果你想设置时间为：2020/01/11 19:09，可选择时间格式为“日期+时分”，再将“自定义格式”处样式设置成“yyyy/MM/dd HH:mm”，效果图如图3.432；
- 如果想让时间的格式为“2019年1月11日 11时11分11秒”，可选择时间格式为“日期+时分秒”，再将“自定义格式”处样式设置成“yyyy年M月dd日hh时mm分ss秒”，效果图如图3.433；
- 如果想让时间的格式为“1月11日”，可选择时间格式为“日期（无年）”，再将自定义格式处样式设置成“M月dd日”，效果图如图3.434；:
- 如果想让时间的格式为“19:12”，可选择时间格式为“时分”，再将自定义格式处样式设置成“HH:mm”，效果图如图3.435；
- 如果想让时间的格式为“19:13:06”，可选择时间格式为“时分秒”，再将自定义格式处样式设置成“HH:mm:ss”，效果图如图3.436；
- 如果想让时间的格式为“星期六”，可选择时间格式为“星期”，再将自定义格式处样式设置成“day”，效果图如图3.437；如果想让时间的格式为“周6”，可在自定义格式处样式设置成“周d”，如图3.438；
- 如果想设置星期，可在“时间格式处”选择“星期”样式；

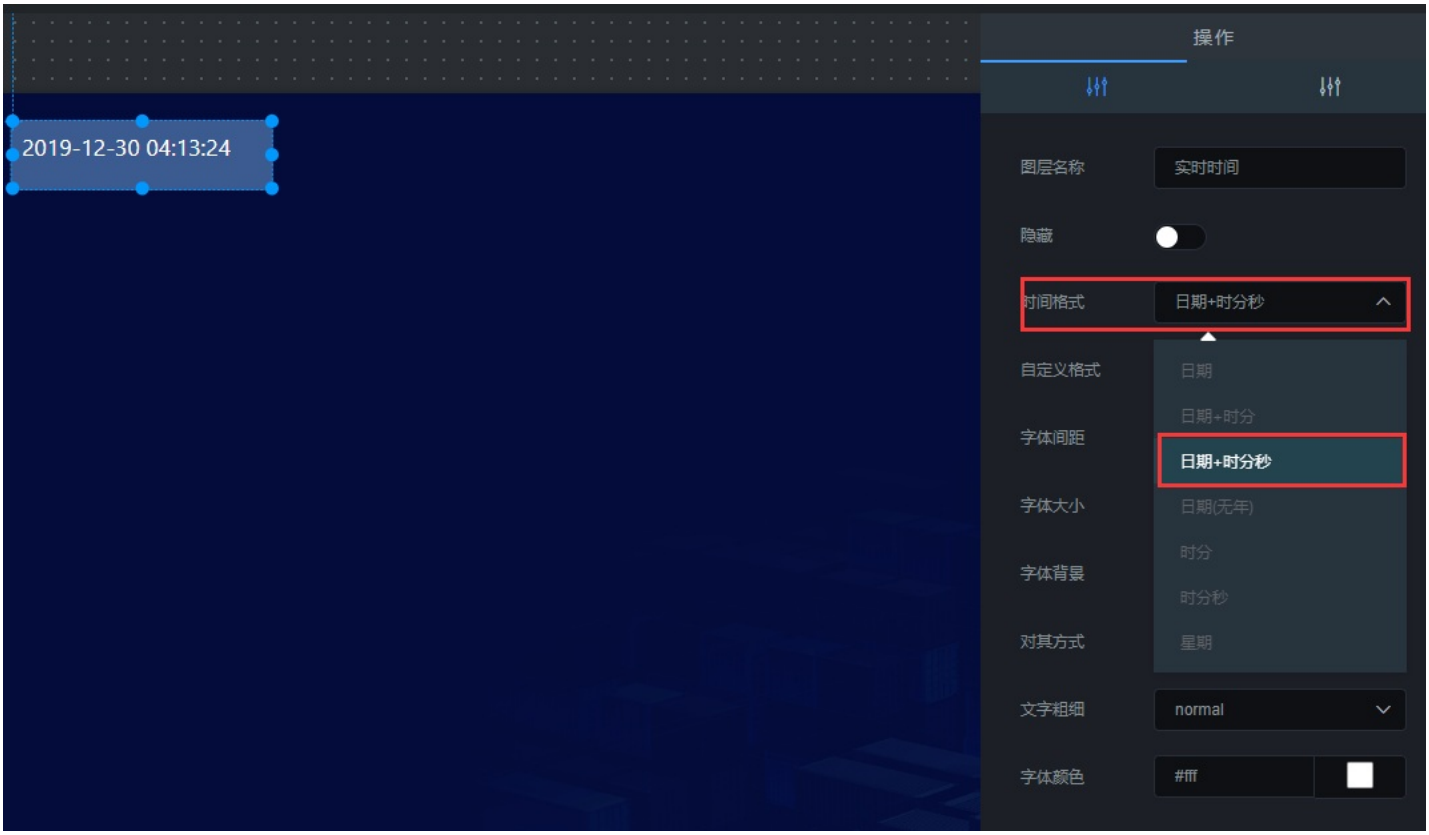


图3. 43

图 3.431 图 3.432 图 3.433 图 3.434

图 3.435 图 3.436 图 3.437 图 3.438

三、文字设置

- 字体间距：文字之间的间距设置；
- 字体大小：文字大小设置；
- 字体背景：文字背景设置；
- 对齐方式：文字对齐方式，包含：居中、左对齐、右对齐；
- 文字粗细：文字的粗细设置；
- 字体颜色：文字颜色设置；

图3. 45

4.图片类组件

4.1图片组件
4.2图片框组件
4.3轮播图组件
4.4滑动组件
4.5iframe组件
4.6video组件

4.1图片组件

图片组件就是设置图片样式的组件。点击“ ”图标，再点击“图片”，即可创建图片，如图4.11；

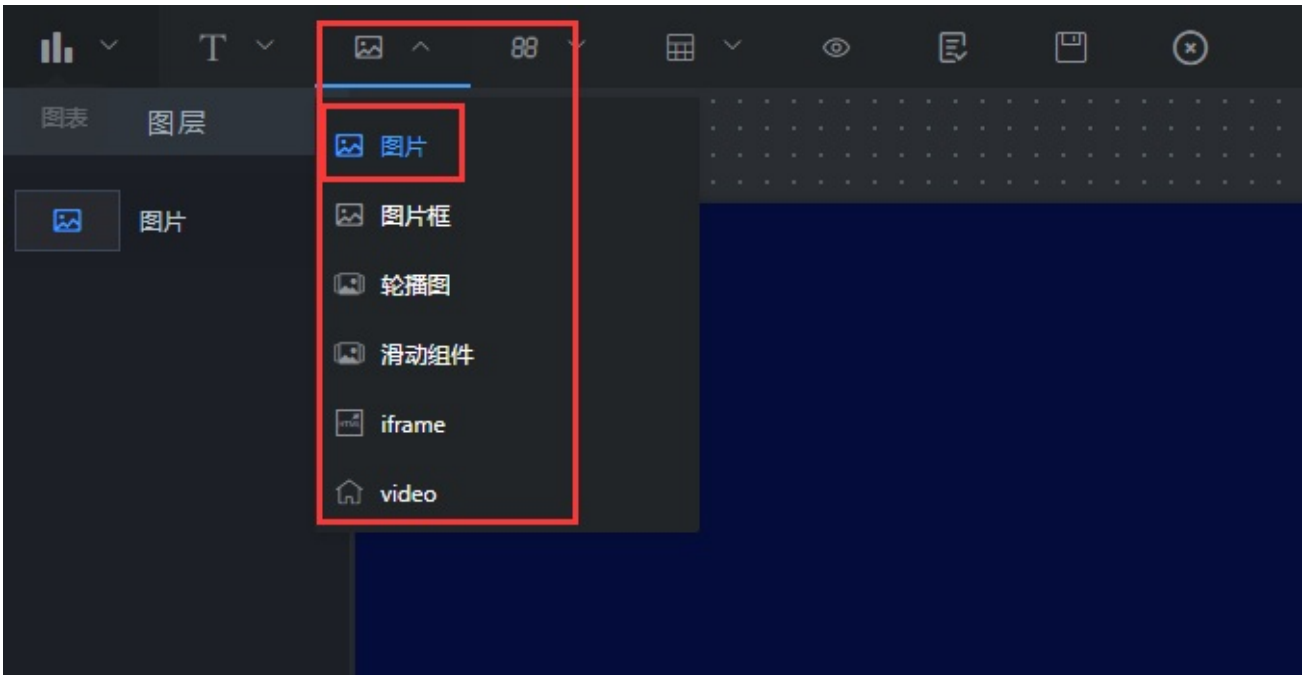


图4. 11

一、组件名称设置

选中该图片组件，在操作界面右侧的“图层名称”处可修改组件的名称，如图4.12。（名称最好要设置一下，方便后期组件管理）

图4. 12

二、图片设置

- 开启旋转：将图片效果设置为旋转，再在“旋转时间”处设置时间，就可控制旋转速度了；
- 透明度：设置图片的透明度；
- 图片地址：可以点击文本框右侧图片按钮，选择图片库内图片；也可在图4.13标注1中填写图片在某一服务器上的存储地址；也可点击图4.13标注2按钮，从本地上传图片；

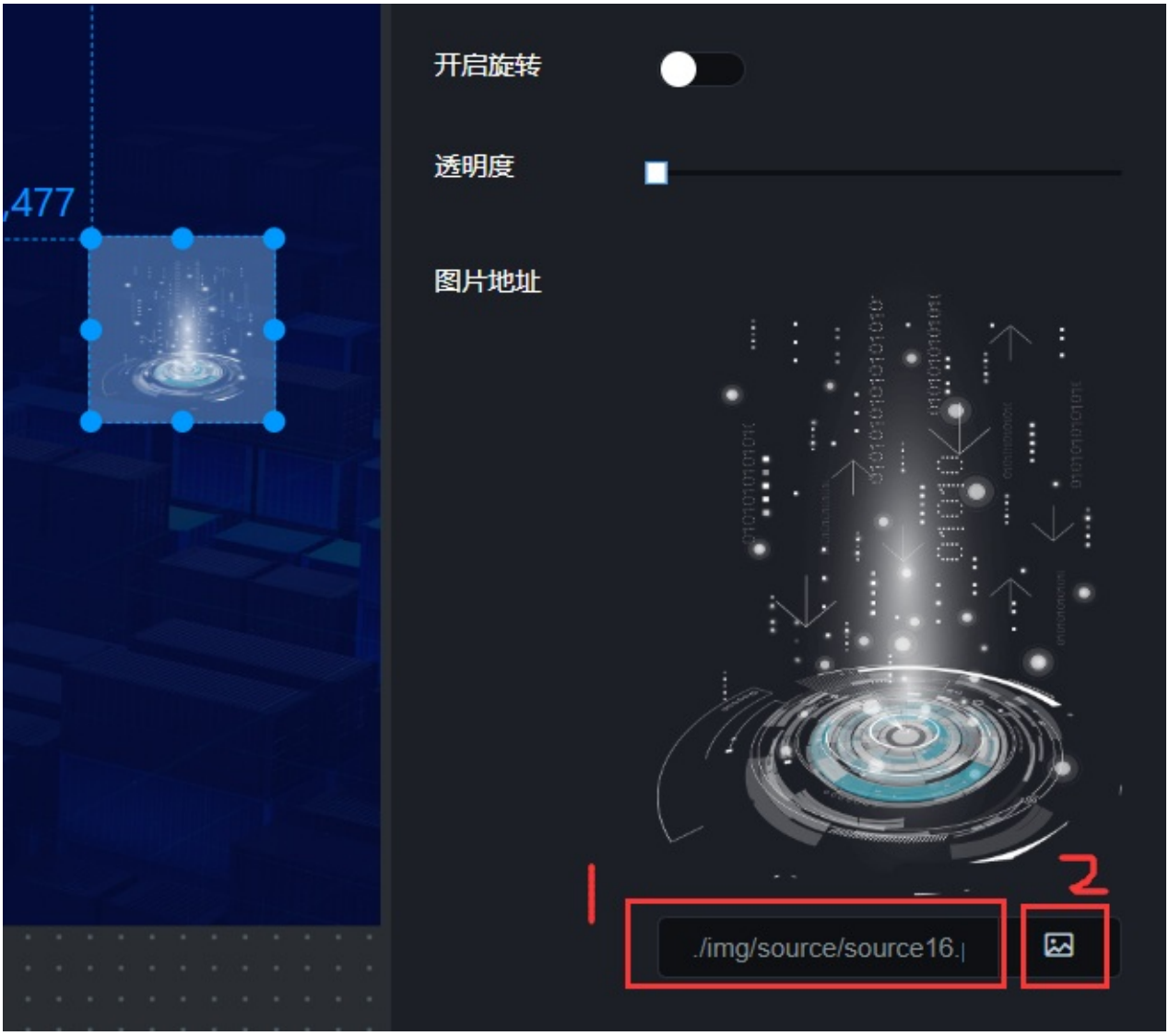


图4. 13

三、接口设置

1. 数据类型

数据类型分为静态数据和动态数据；

- 静态数据：写死的数据；
- 动态数据：会随着接口传过来的数据实时变化；（一般这种比较常用）

2. 接口地址

（1）静态数据，接口地址传过来的内容要符合以下格式：

```
{
    "value": "https://img.alicdn.com/tfs/TB1v28TC8v0gK0jSZKbXXbK2FXa-1880-640.jpg"
}
```

（2）动态数据，接口地址传过来的内容要符合以下格式：

```
{"value":"https://img.alicdn.com/tfs/TB1v28TC8v0gK0jSZKbXXbK2FXa-1880-640.jpg"}
```

3. 刷新时间

这个参数主要针对动态数据设置的，完成数据的实时更新。

- 如果你想设置成5秒刷新一次，可以将刷新时间设置成“5000”；

4. 刷新数据

这个参数主要是重新请求以下接口，完成数据的更新。


操作
数据类型
静态数据
动态数据
接口地址
/mock/26/img
接口方式
POST
GET
接口参数
编辑
刷新时间
数据处理
编辑
刷新
,326
激荡2020

图4. 14

4.2图片框组件

图片框组件就是设置图片的边框样式。点击“ ”图标，再点击“图片框”，即可创建图片框，如图4.21；

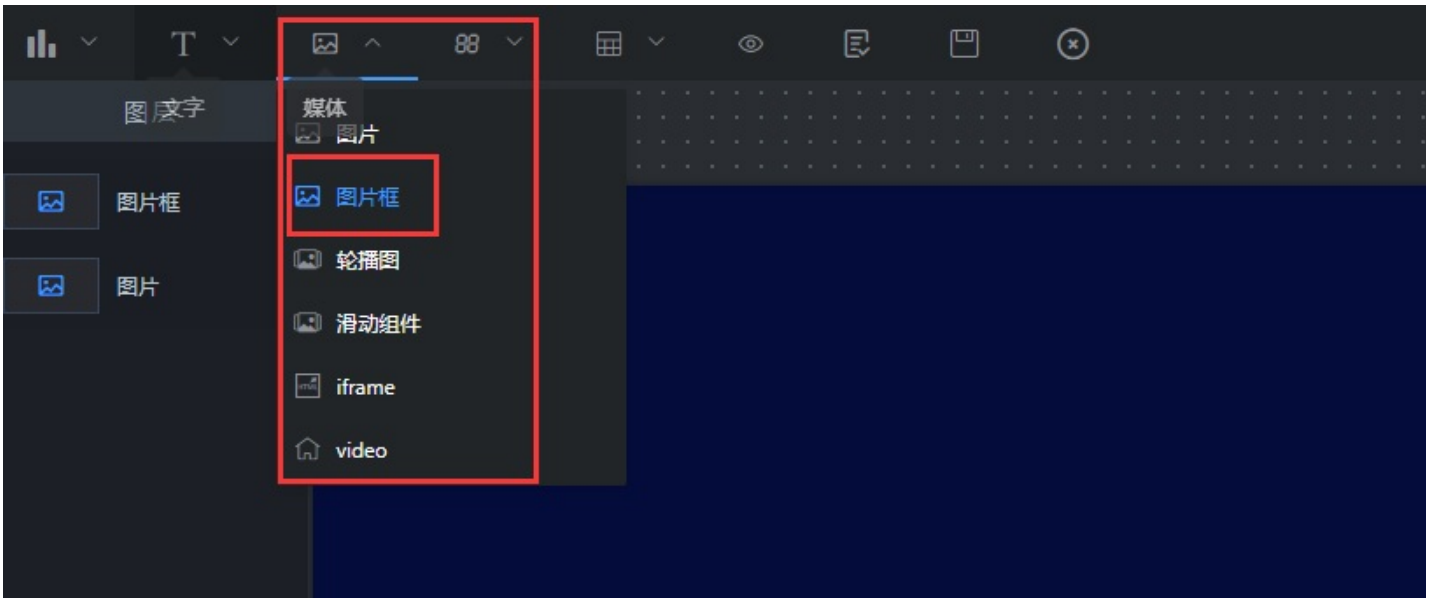


图4. 21

一、组件名称设置

选中该图片框组件，在操作界面右侧的“图层名称”处可修改组件的名称，如图4.22。（名称最好要设置一下，方便后期组件管理）

图4. 22

二、图片设置

- 背景色：设置图片框的纯色背景；（如果想要显示背景颜色，需要先清空一下图片地址）
- 图片透明度：设置图片的透明程度；
- 图片地址：可以点击文本框右侧图片按钮，选择图片库内图片；也可在图4.23标注1中填写图片在某一服务器上的存储地址；也可点击图4.23标注2按钮，从本地上传图片；

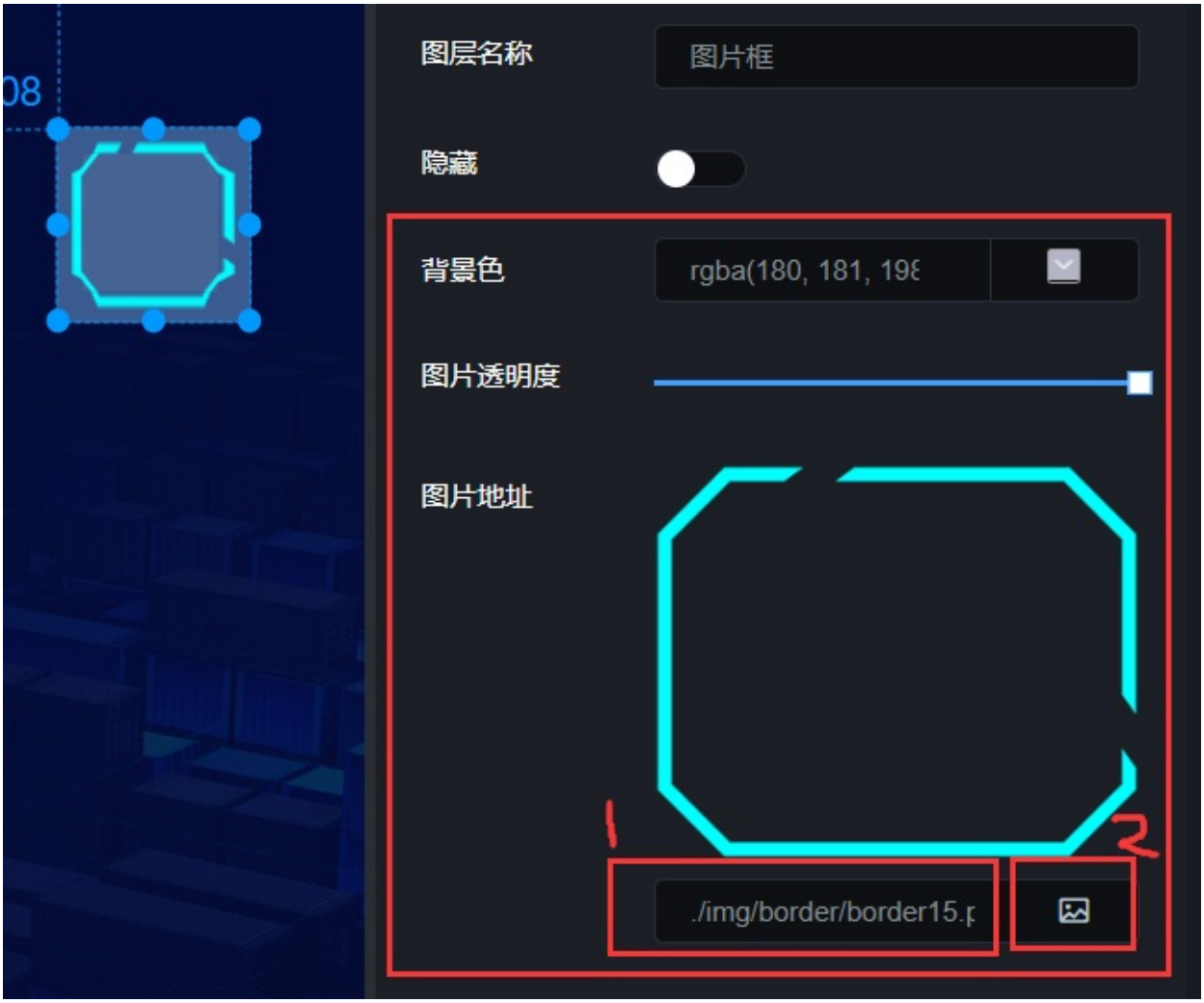

图层名称
图片框
隐藏
背景色
rgba(180, 181, 198
图片透明度
图片地址
./img/border/border15.p
1
2

图4. 23

4.3轮播图组件

轮播图组件就是设置图片滚动效果的组件。点击“ ”图标，再点击“轮播图”，即可创建轮播图，如图4.31；

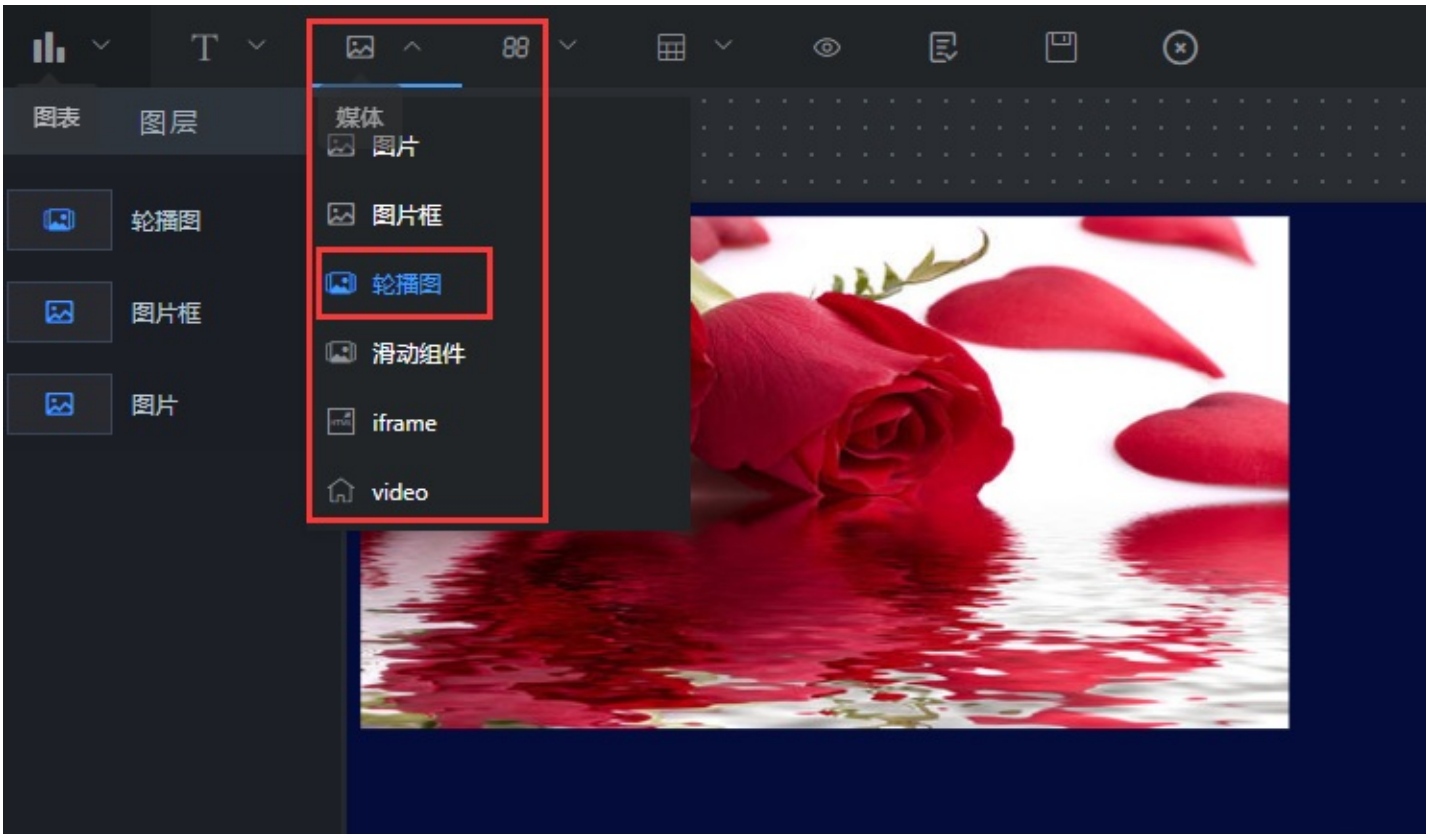


图4. 31

一、组件名称设置

选中该轮播图组件，在操作界面右侧的“图层名称”处可修改组件的名称，如图4.32。（名称最好要设置一下，方便后期组件管理）

图4. 32

二、基本样式设置

1. 图片类型

选中该轮播图组件，在操作界面右侧的“类型”处可设置组件的类型，如图4.33；图片类型分为：立体和普通；

- 普通：我们常见的轮播图样式，效果图如图4.34；
- 立体：立体效果的轮播图，效果图如图4.35；

图4. 33

图4. 34

图4. 35

2. 轮播时间

选中该轮播图组件，在操作界面右侧的“轮播时间”处可设置组件的轮播时间，如图4.36；

- 如果你想让图片5秒切换到下一张，将轮播时间设置成“5000”；

图4. 36

3. 选择器

选中该轮播图组件，在操作界面右侧的“选择器”处可设置组件底部提示线的显示与否，如图4.37。

- 外部：组件底部提示线显示；
- 不显示：组件底部提示线不显示；

图4. 37

4. 轮播方向

选中该轮播图组件，在操作界面右侧的“轮播方向”处可设置轮播图的轮播方向，如图4.38。

- 竖向：竖着轮播，如图4.381；
- 横向：横着轮播，图4.382；

图4. 38

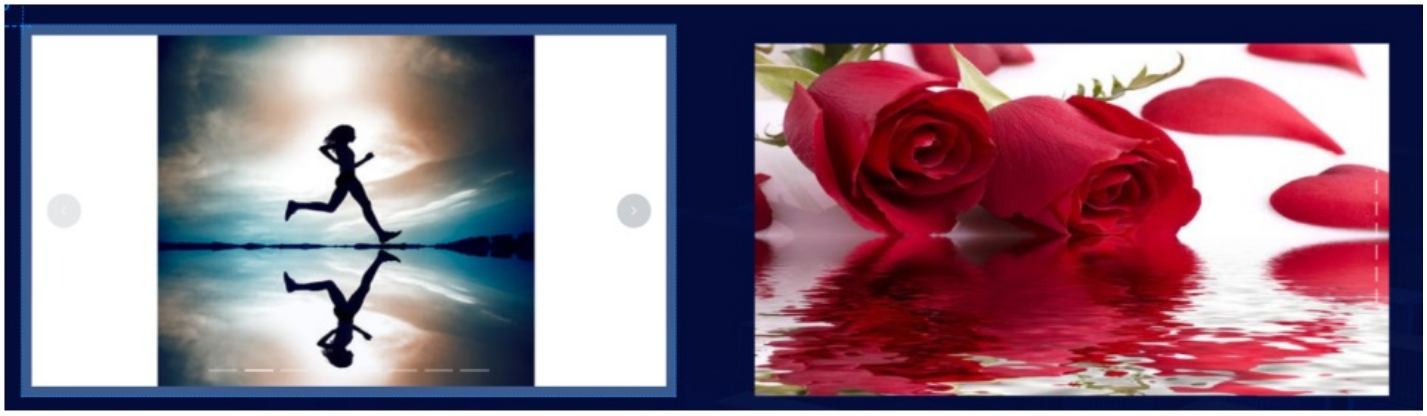


图 4.381

图 4.382

5. 图片透明度

选中该轮播图组件，在操作界面右侧的“透明度”处可设置图片透明度，如图4.39。

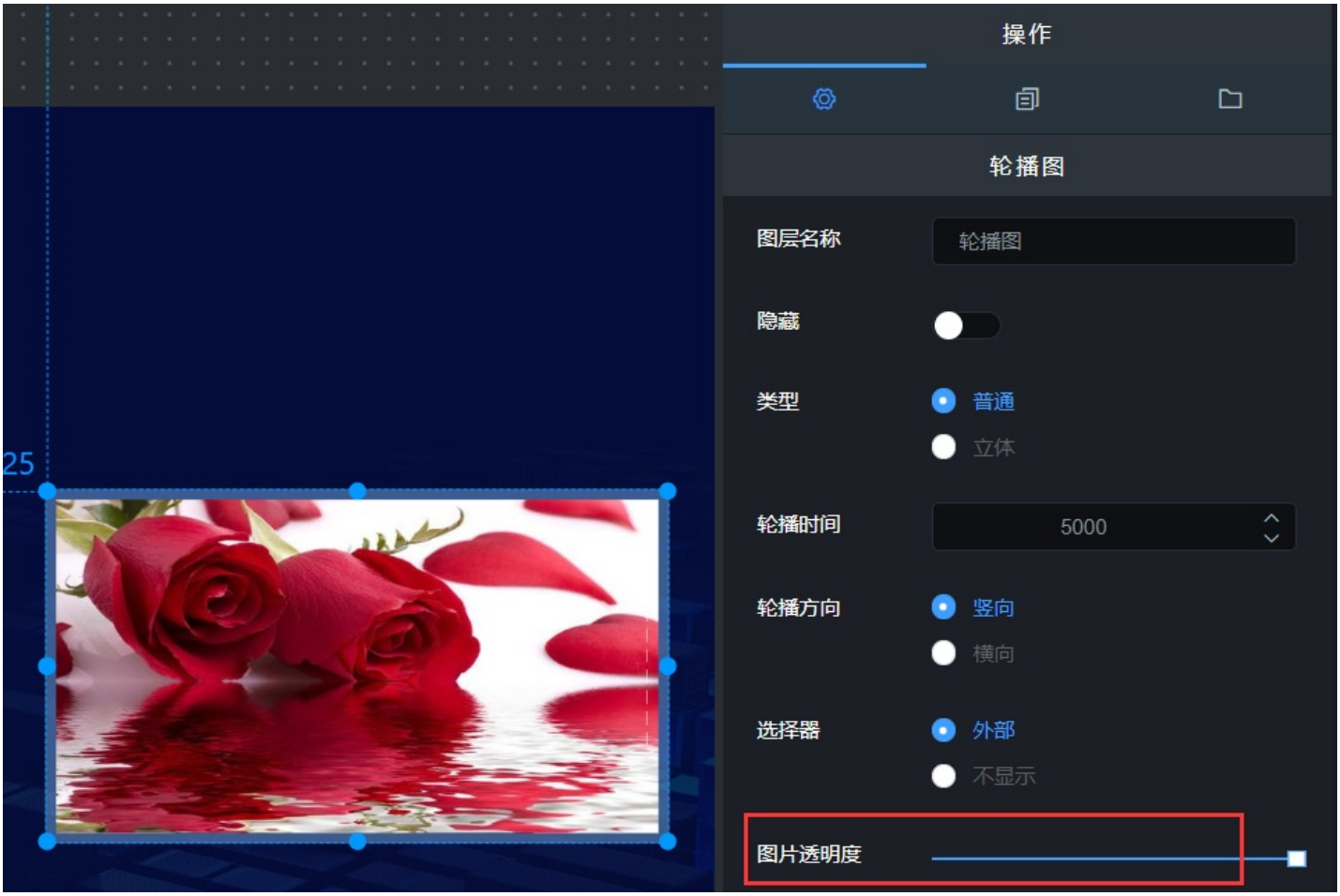


图4. 39

三、接口设置

1. 数据类型

数据类型分为静态数据和动态数据；

- 静态数据：写死的数据；
- 动态数据：会随着接口传过来的数据实时变化；（一般这种比较常用）

2. 接口地址

（1）静态数据，接口地址传过来的内容要符合以下格式：

```
[{"value":"https://dss0.bdstatic.com/70cFuHSh_Q1YnxGkpoWK1HF6hhy/it/u=2229864841,4232235061&fm=26&gp=0.jpg"},{"value":"https://dss1.bdstatic.com/70cFvXSh_Q1YnxGkpoWK1HF6hhy/it/u=4238142487,3274484296&fm=26&gp=0.jpg"},{"value":"https://dss0.bdstatic.com/70cFvHSh_Q1YnxGkpoWK1HF6hhy/it/u=2394972844,3024358326&fm=26&gp=0.jpg"},{"value":"https://dss0.bdstatic.com/70cFuHSh_Q1YnxGkpoWK1HF6hhy/it/u=2229864841,4232235061&fm=26&gp=0.jpg"},{"value":"https://dss1.bdstatic.com/70cFvXSh_Q1YnxGkpoWK1HF6hhy/it/u=4238142487,3274484296&fm=26&gp=0.jpg"},{"value":"https://dss0.bdstatic.com/70cFvHSh_Q1YnxGkpoWK1HF6hhy/it/u=2394972844,3024358326&fm=26&gp=0.jpg"},{"value":"https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRVq8ZVBlQLcZMHXjm21FMbF4oMmyCucZZav2s_wcd_Wkosp4060g&s"},{"value":"https://en
```

```
crypted-tbn0.gstatic.com/images?q=tbn:ANd9GcS4QJcBTHThqzbA9Of6HzW2i1apf5k2i2iIa
QQPLR0UT2AOH7ke&s"}]
```

（2）动态数据，接口地址传过来的内容要符合以下格式：

```
{"data":[{"value":"https://dss0.bdstatic.com/70cFuHSh_Q1YnxGkpoWK1HF6hhy/it/u=2
229864841,4232235061&fm=26&gp=0.jpg"},{"value":"https://dss1.bdstatic.com/70cFv
XSh_Q1YnxGkpoWK1HF6hhy/it/u=4238142487,3274484296&fm=26&gp=0.jpg"},{"value":"ht
tps://dss0.bdstatic.com/70cFvHSh_Q1YnxGkpoWK1HF6hhy/it/u=2394972844,3024358326&
fm=26&gp=0.jpg"},{"value":"https://dss0.bdstatic.com/70cFuHSh_Q1YnxGkpoWK1HF6hh
y/it/u=2229864841,4232235061&fm=26&gp=0.jpg"},{"value":"https://dss1.bdstatic.c
om/70cFvXSh_Q1YnxGkpoWK1HF6hhy/it/u=4238142487,3274484296&fm=26&gp=0.jpg"},{"va
lue":"https://dss0.bdstatic.com/70cFvHSh_Q1YnxGkpoWK1HF6hhy/it/u=2394972844,302
4358326&fm=26&gp=0.jpg"},{"value":"https://encrypted-tbn0.gstatic.com/images?q=
tbn:ANd9GcRVq8ZVBlQLcZMHXjm21FMbF4oMmyCucZZav2s_wcd_Wkosp4060g&s"},{"value":"ht
tps://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcS4QJcBTHThqzbA9Of6HzW2i1apf
5k2i2iIaQQPLR0UT2AOH7ke&s"}]}
```

3. 刷新时间

这个参数主要针对动态数据设置的，完成数据的实时更新。

- 如果你想设置成5秒刷新一次，可以将刷新时间设置成“5000”；

4. 刷新数据

这个参数主要是重新请求以下接口，完成数据的更新。

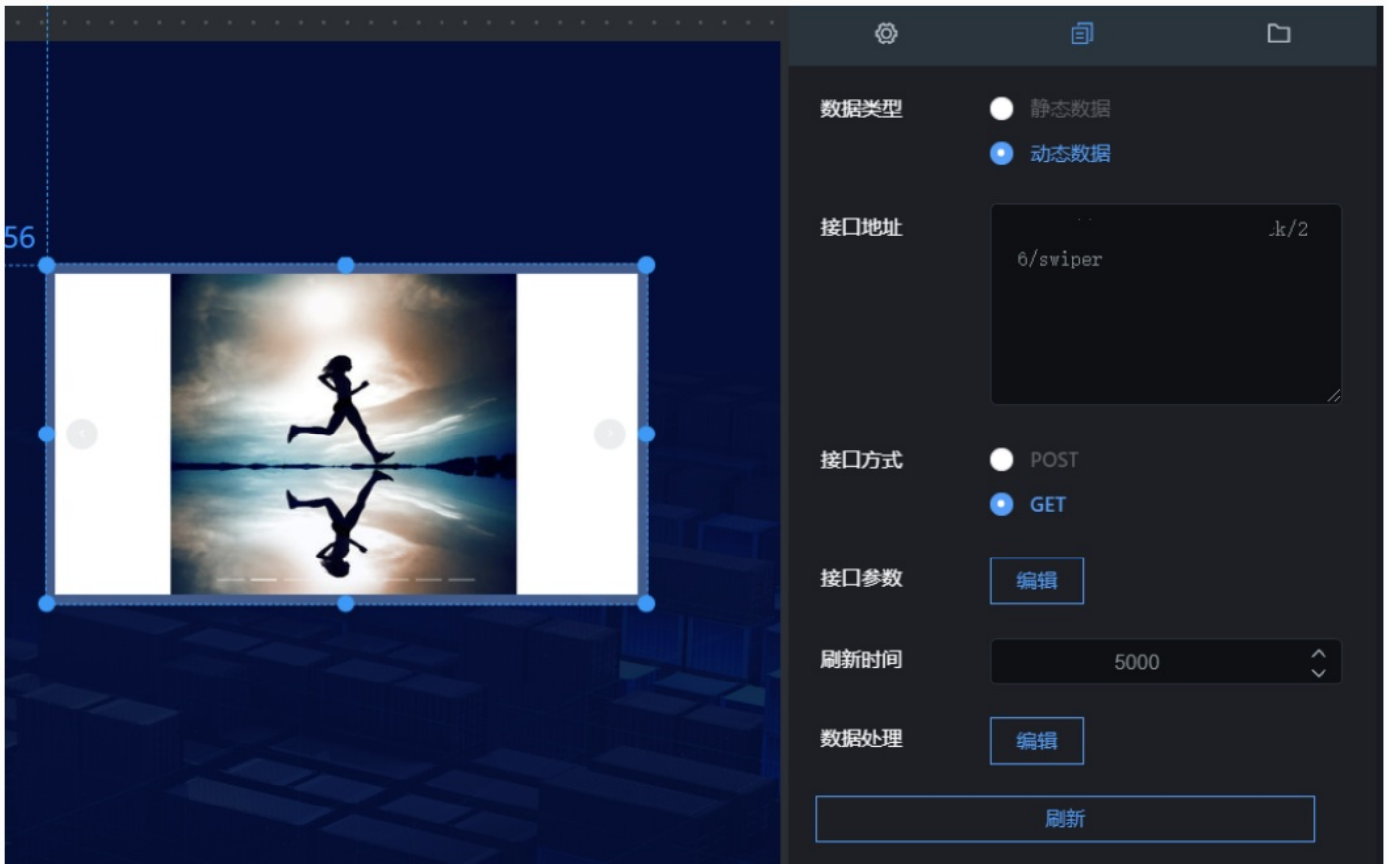

数据类型
静态数据
动态数据
接口地址
6/swiper
接口方式
POST
GET
接口参数
编辑
刷新时间
5000
数据处理
编辑
刷新

图4.39

4.4滑动组件

滑动组件就是设置滑动样式的组件。点击“ ”图标，再点击“滑动组件”，即可创建新的滑动，如图4.41；

图4.41

一、组件名称设置

选中该滑动组件，在操作界面右侧的“图层名称”处可修改组件的名称，如图4.42。（名称最好要设置一下，方便后期组件管理）

图4.42

二、接口设置

1. 数据类型

数据类型分为静态数据和动态数据；

- 静态数据：写死的数据；
- 动态数据：会随着接口传过来的数据实时变化；（一般这种比较常用）

2. 接口地址

（1）静态数据，接口地址传过来的内容要符合以下格式：

```
[{"value":"https://dss0.bdstatic.com/70cFuHSh_Q1YnxGkpoWK1HF6hhy/it/u=2229864841,4232235061&fm=26&gp=0.jpg"},{"value":"https://dss1.bdstatic.com/70cFvXSh_Q1YnxGkpoWK1HF6hhy/it/u=4238142487,3274484296&fm=26&gp=0.jpg"},{"value":"https://dss0.bdstatic.com/70cFvHSh_Q1YnxGkpoWK1HF6hhy/it/u=2394972844,3024358326&fm=26&gp=0.jpg"},{"value":"https://dss0.bdstatic.com/70cFuHSh_Q1YnxGkpoWK1HF6hhy/it/u=2229864841,4232235061&fm=26&gp=0.jpg"},{"value":"https://dss1.bdstatic.com/70cFvXSh_Q1YnxGkpoWK1HF6hhy/it/u=4238142487,3274484296&fm=26&gp=0.jpg"},{"value":"https://dss0.bdstatic.com/70cFvHSh_Q1YnxGkpoWK1HF6hhy/it/u=2394972844,3024358326&fm=26&gp=0.jpg"},{"value":"https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9
```

```
GcRVq8ZVBlQLcZMHXjm21FMbF4oMmyCucZZav2s_wcd_Wkosp4060g&s"},{"value":"https://en
crypted-tbn0.gstatic.com/images?q=tbn:ANd9GcS4QJcBTHThqzbA9Of6HzW2i1apf5k2i2iIa
QQPLR0UT2AOH7ke&s"}]
```

（2）动态数据，接口地址传过来的内容要符合以下格式：

```
{"data":[{"value":"https://dss0.bdstatic.com/70cFuHSh_Q1YnxGkpoWK1HF6hhy/it/u=2
229864841,4232235061&fm=26&gp=0.jpg"},{"value":"https://dss1.bdstatic.com/70cFv
XSh_Q1YnxGkpoWK1HF6hhy/it/u=4238142487,3274484296&fm=26&gp=0.jpg"},{"value":"ht
tps://dss0.bdstatic.com/70cFvHSh_Q1YnxGkpoWK1HF6hhy/it/u=2394972844,3024358326&
fm=26&gp=0.jpg"},{"value":"https://dss0.bdstatic.com/70cFuHSh_Q1YnxGkpoWK1HF6hh
y/it/u=2229864841,4232235061&fm=26&gp=0.jpg"},{"value":"https://dss1.bdstatic.c
om/70cFvXSh_Q1YnxGkpoWK1HF6hhy/it/u=4238142487,3274484296&fm=26&gp=0.jpg"},{"va
lue":"https://dss0.bdstatic.com/70cFvHSh_Q1YnxGkpoWK1HF6hhy/it/u=2394972844,302
4358326&fm=26&gp=0.jpg"},{"value":"https://encrypted-tbn0.gstatic.com/images?q=
tbn:ANd9GcRVq8ZVBlQLcZMHXjm21FMbF4oMmyCucZZav2s_wcd_Wkosp4060g&s"},{"value":"ht
tps://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcS4QJcBTHThqzbA9Of6HzW2i1apf
5k2i2iIaQQPLR0UT2AOH7ke&s"}]}
```

3. 刷新时间

这个参数主要针对动态数据设置的，完成数据的实时更新。

- 如果你想设置成5秒刷新一次，可以将刷新时间设置成“5000”；

4. 刷新数据

这个参数主要是重新请求以下接口，完成数据的更新。


操作
数据类型
静态数据
动态数据
接口地址
6/swiper
接口方式
POST
GET
接口参数
编辑
刷新时间
5000
数据处理
编辑
刷新
分娩舍
育成舍1
yuchengshe1
育成舍2
yuchengshe2
育肥舍
yufeishe

图4.43

4.5iframe组件

iframe组件就是设置框架的组件。点击“ ”图标，再点击“iframe” ，即可创建iframe，如图4.51；

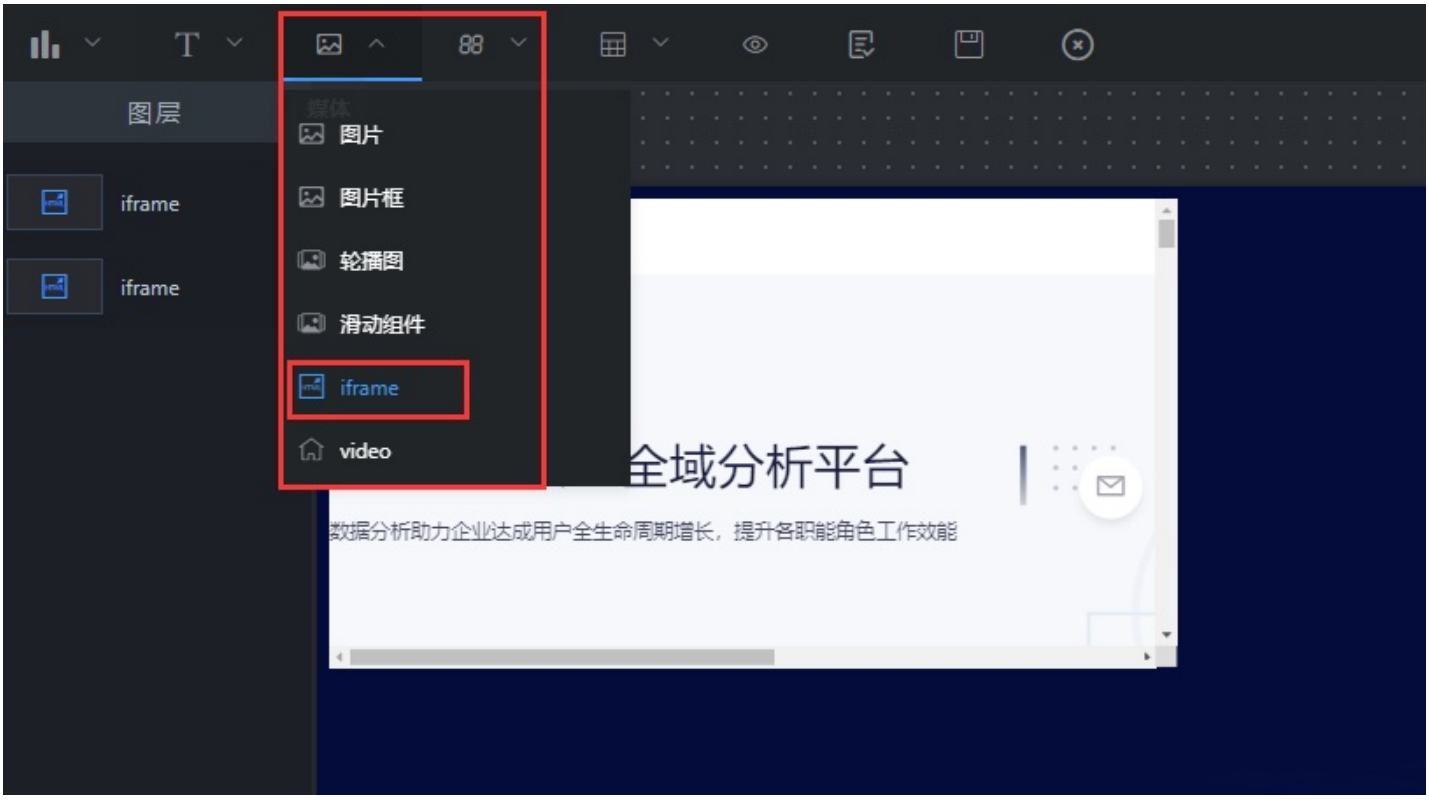


图4.51

一、组件名称设置

选中该 iframe组件，在操作界面右侧的“图层名称”处可修改组件的名称，如图4.52。（名称最好要设置一下，方便后期组件管理）

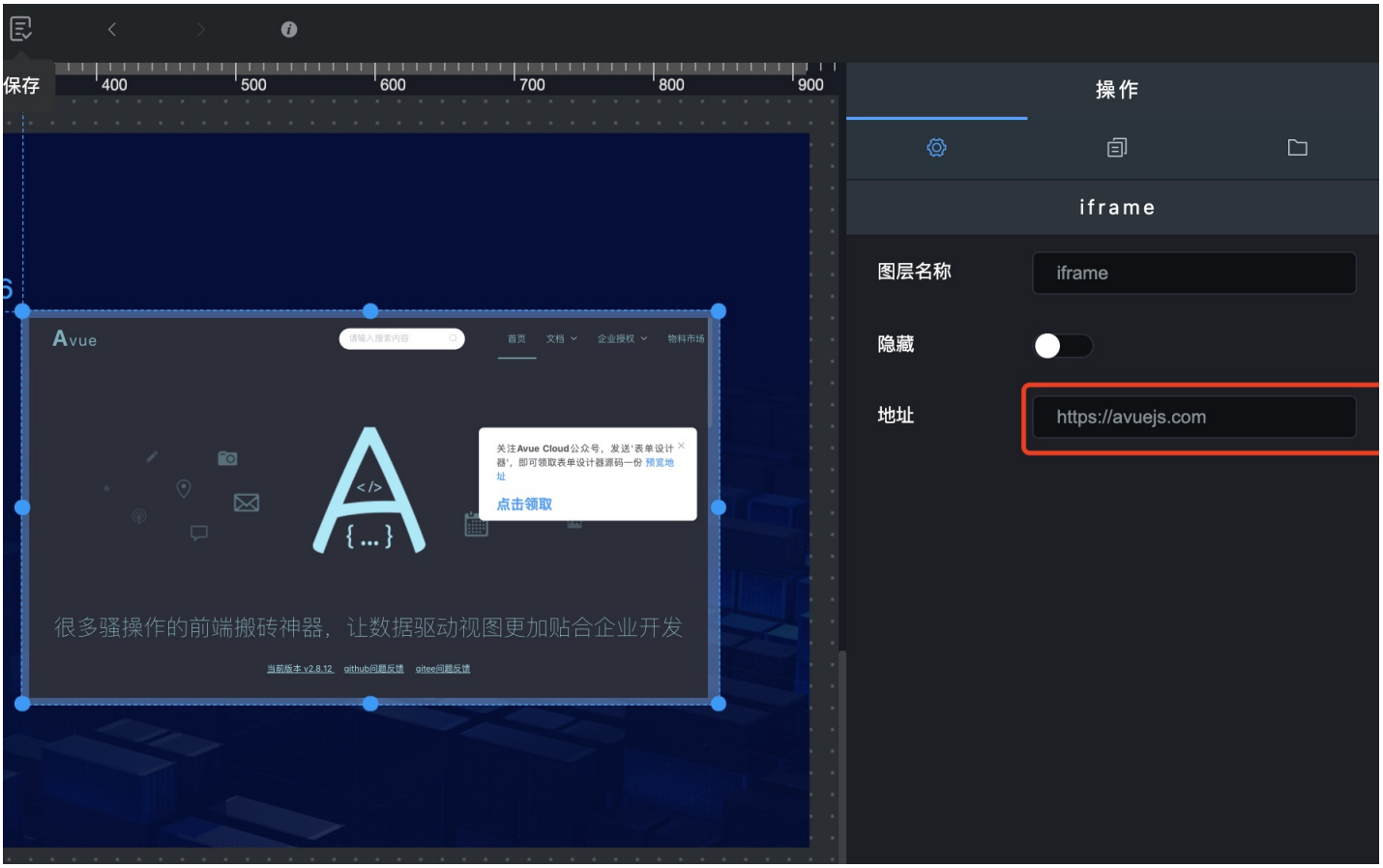


图4.52

二、地址

该iframe组件显示的地址；

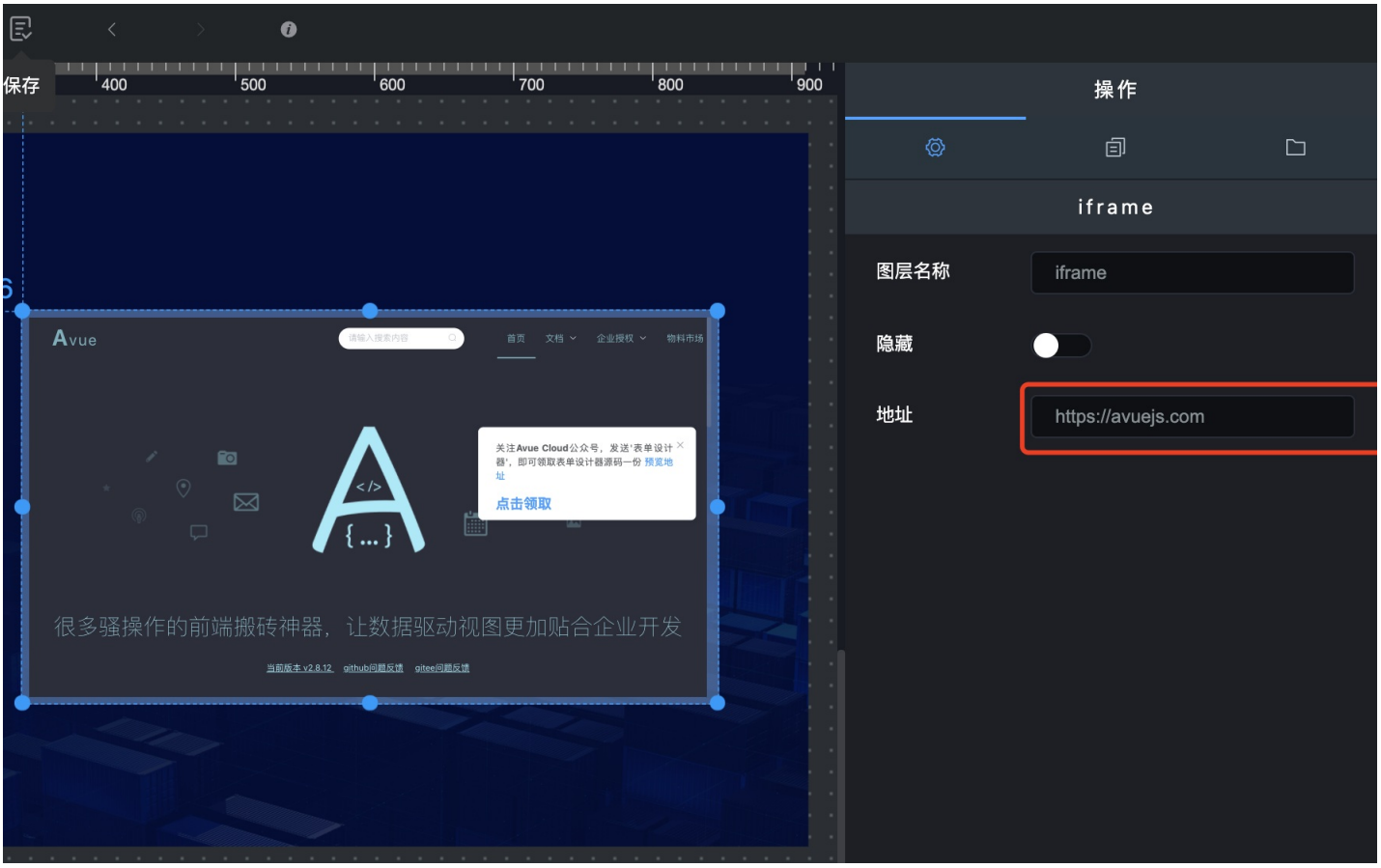


图4.53

三、接口设置

1. 数据类型

数据类型分为静态数据和动态数据；

- 静态数据：写死的数据；
- 动态数据：会随着接口传过来的数据实时变化；（一般这种比较常用）

2. 接口地址

（1）静态数据，接口地址传过来的内容要符合以下格式：

```
{
    "value": "https://cloud.baidu.com"
}
```

（2）动态数据，接口地址传过来的内容要符合以下格式：

```
{"data":{"value":"http://news.baidu.com/"}}
```

3. 刷新时间

这个参数主要针对动态数据设置的，完成数据的实时更新。

- 如果你想设置成5秒刷新一次，可以将刷新时间设置成“5000”；

4. 刷新数据

这个参数主要是重新请求以下接口，完成数据的更新。

4.6video组件

video组件就是添加音/视频的组件。点击“ ”图标，再点击“video” ，即可创建音/视频，如图4.61；

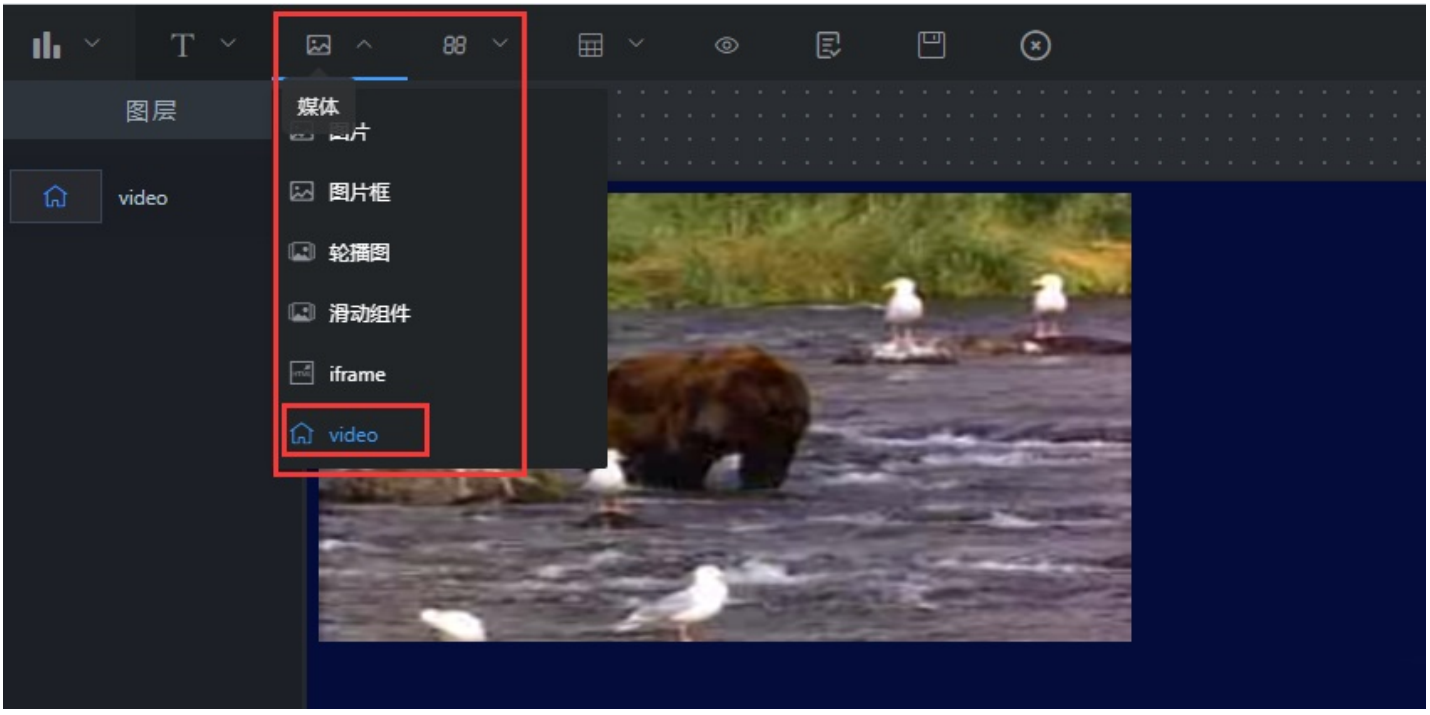


图4.61

一、组件名称设置

选中该video组件，在操作界面右侧的“图层名称”处可修改组件的名称，如图4.62。（名称最好要设置一下，方便后期组件管理）

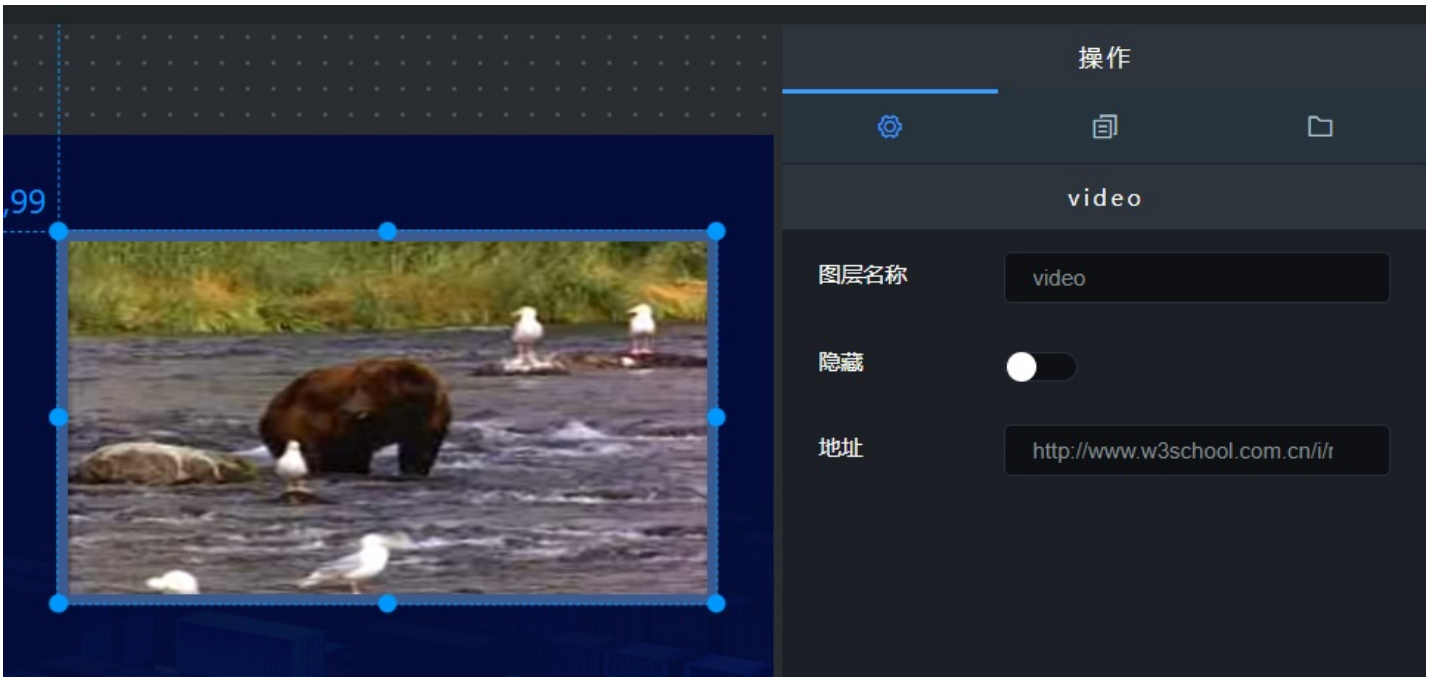


图4. 62

二、地址

该video组件显示的地址；

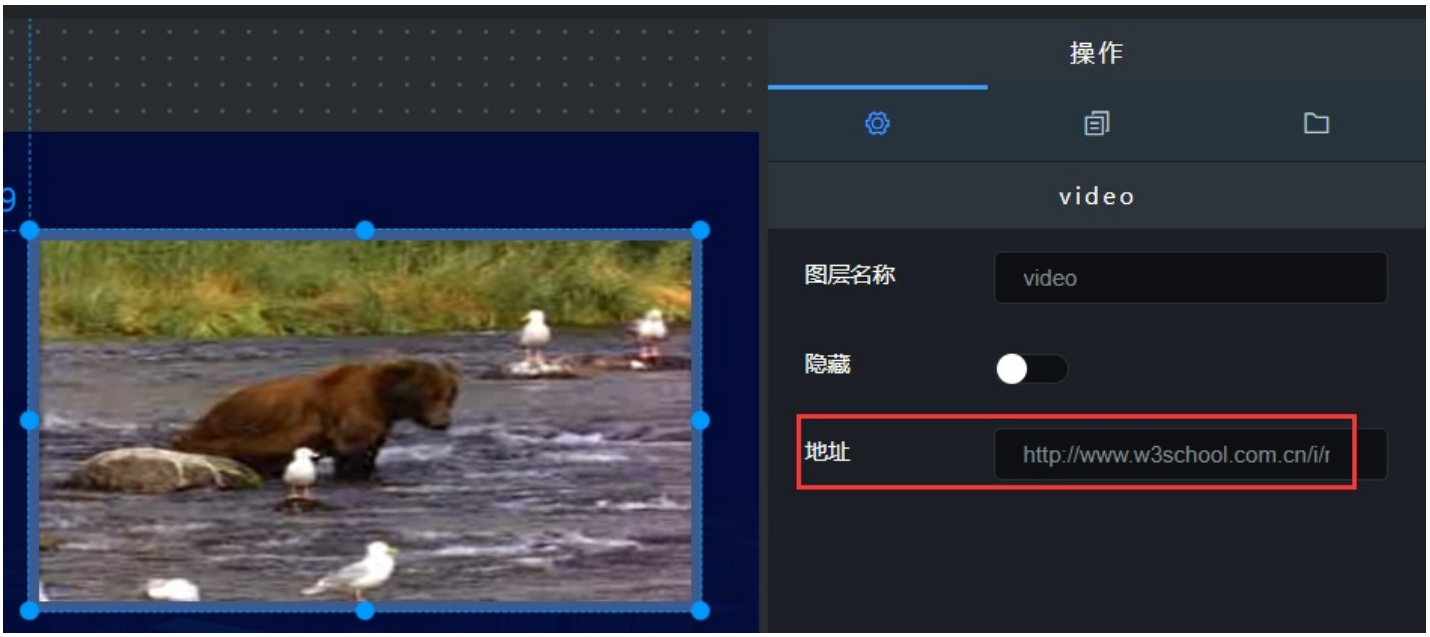


图4. 63

三、接口设置

1. 数据类型

数据类型分为静态数据和动态数据；

- 静态数据：写死的数据；
- 动态数据：会随着接口传过来的数据实时变化；（一般这种比较常用）

2. 接口地址

（1）静态数据还是动态数据，接口地址穿过来的内容要符合以下格式：

```
{
    "value": "http://www.w3school.com.cn/i/movie.ogg"
}
```

（2）动态数据还是动态数据，接口地址穿过来的内容要符合以下格式：

```
{"data":{"value":"https://elekta.cn/fileadmin/groups/editors/videos/Elekta_Brand_Film_small.mp4"}}
```

3. 刷新时间

这个参数主要针对动态数据设置的，完成数据的实时更新。

- 如果你想设置成5秒刷新一次，可以将刷新时间设置成“5000”；

4. 刷新数据

这个参数主要是重新请求以下接口，完成数据的更新。

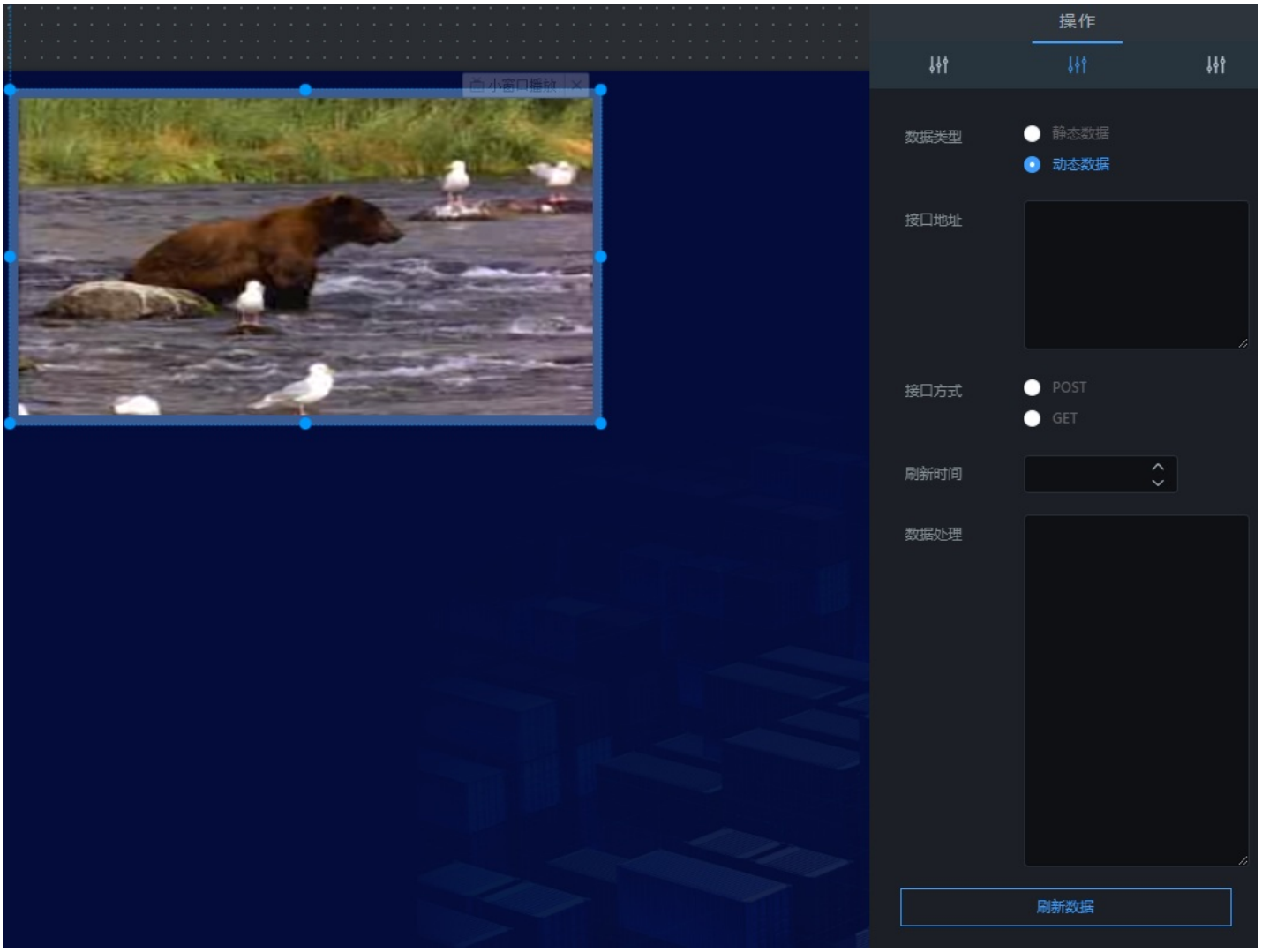


图4. 63

5.指标类组件

5.1翻牌器组件

5.2仪盘表组件

5.3字符云组件

5.5进度条组件

5.1翻牌器组件

翻牌器组件就是添加数字各种样式的组件。点击“ 88 ”图标，再点击“翻牌器”，即可创翻牌器，如图5.11；

图5.11

一、组件名称设置

选中该翻牌器组件，在操作界面右侧的“图层名称”处可修改组件的名称，如图5.12。（名称最好要设置一下，方便后期组件管理）

图5. 12

二、基本样式设置

1. 整体

选中该字符云组件，在操作界面右侧的“整体”处可把数字设置成整体的样式，如图5.13；

图5. 13

2. 横行

选中该字符云组件，在操作界面右侧的“横行”处可对横行 进行设置样式，如图5.14；

图5. 14

3. 精度

选中该字符云组件，在操作界面右侧的“精度”处修改，可设置数值保留的小数位数，设置好后，点击预览可以看到效果，如图5.15，预览样式如图5.151；

图5.15

图5.151

4. 边框设置

选中该字符云组件，在操作界面右侧的“边框设置”处修改，设置好后，点击预览可以看到效果，如图5.16；

- 间距：边款与文字的间距，默认文字居中，调整间距值，文字向下移动；
- 边框：可设置边框样式；分为：无边框、内置图片、内置边框；

```
* 无边框：设置无边框的翻牌器，可以用这个设置一些大一点的数字或者钱，样式如图5.161；
* 内置图片：设置翻牌器的背景，如可设置图片地址、背景图片、背景颜色，样式如图5.162；
* 内置边框：设置翻牌器的边框样式，如可设置边框颜色、边框宽度、背景颜色，样式如图5.163
```

;

图5. 16

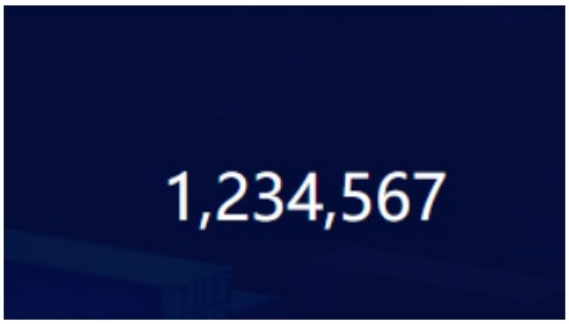


图 5.161

图 5.162

图 5.163

5. 内部设置

选中该字符云组件，在操作界面右侧的“内部设置”处可修改文字样式，如图5.17；

* 字体大小：翻牌器内文字的大小；
* 字体颜色：翻牌器内文字颜色；
* 字体粗细：翻牌器内文字粗细；
* 对齐方式：翻牌器内文字对齐方式；

图5. 17

6. 前缀设置

选中该字符云组件，在操作界面右侧的“前缀设置”处可设置前缀样式，如图5.18；

* 前缀内容：前缀要显示的内容；
* 对齐方式：前缀的对齐方式；
* X间距：前缀跟X轴的间距；
* Y间距：前缀跟Y轴间距；
* 颜色：前缀字体颜色；
* 字体大小：前缀字体大小；

图5.18

7. 后缀设置

选中该字符云组件，在操作界面右侧的“前缀设置”处可设置前缀样式，如图5.19；

* 前缀内容：前缀要显示的内容；
* 对齐方式：前缀的对齐方式；
* X间距：前缀跟X轴的间距；
* Y间距：前缀跟Y轴间距；
* 颜色：前缀字体颜色；
* 字体大小：前缀字体大小；

图5.19

三、接口设置

1. 数据类型

数据类型分为静态数据和动态数据；

- 静态数据：写死的数据；
- 动态数据：会随着接口传过来的数据实时变化；（一般这种比较常用）

2. 接口地址

（1）静态数据，接口地址穿过来的内容要符合以下格式：

```
{
    "value": "12345"
}
```

（2）动态数据，接口地址穿过来的内容要符合以下格式：

```
{"data":{"value":"1234567"}}
```

3. 刷新时间

这个参数主要针对动态数据设置的，完成数据的实时更新。

- 如果你想设置成5秒刷新一次，可以将刷新时间设置成“5000”；

4. 刷新数据

这个参数主要是重新请求以下接口，完成数据的更新。


操作
数据类型
静态数据
动态数据
接口地址
6/flop
12,345.00
接口方式
POST
GET
接口参数
编辑
刷新时间
3000
数据处理
编辑
刷新

图1.45

5.2仪盘表组件

仪盘表组件就是设置仪盘效果的组件，效果图如图5.21。点击“ ”图标，再点击“仪盘表” ，即可创建仪盘表，如图5.22；

图5. 21

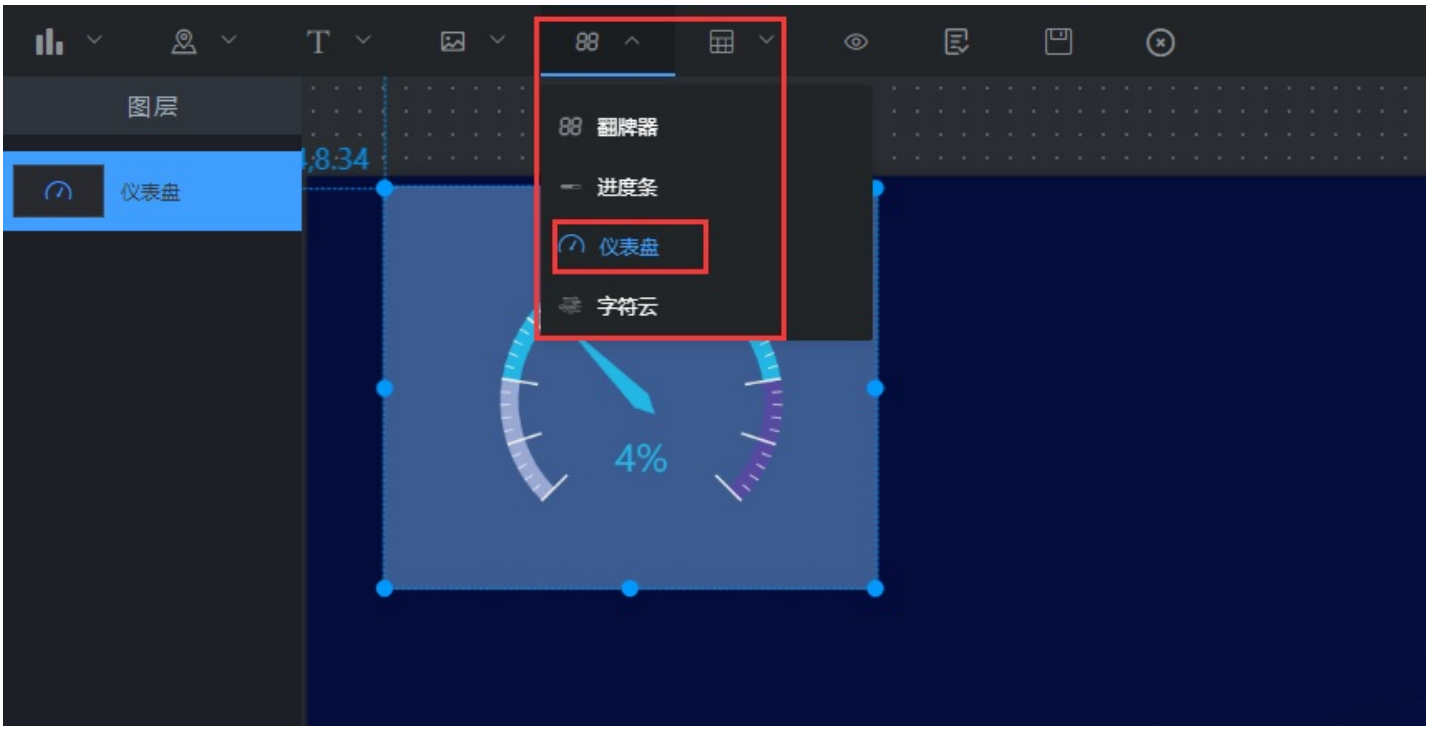


图5. 22

一、组件名称设置

选中该仪盘表组件，在操作界面右侧的“图层名称”处可修改组件的名称，如图5.23。（名称最好要设置一下，方便后期组件管理）

图5. 23

二、基本样式设置

1. 刻度线粗度(像素)

选中该仪盘表组件，在操作界面右侧的“刻度线粗度(像素)”处可设置组件的轮播时间，如图5.24；

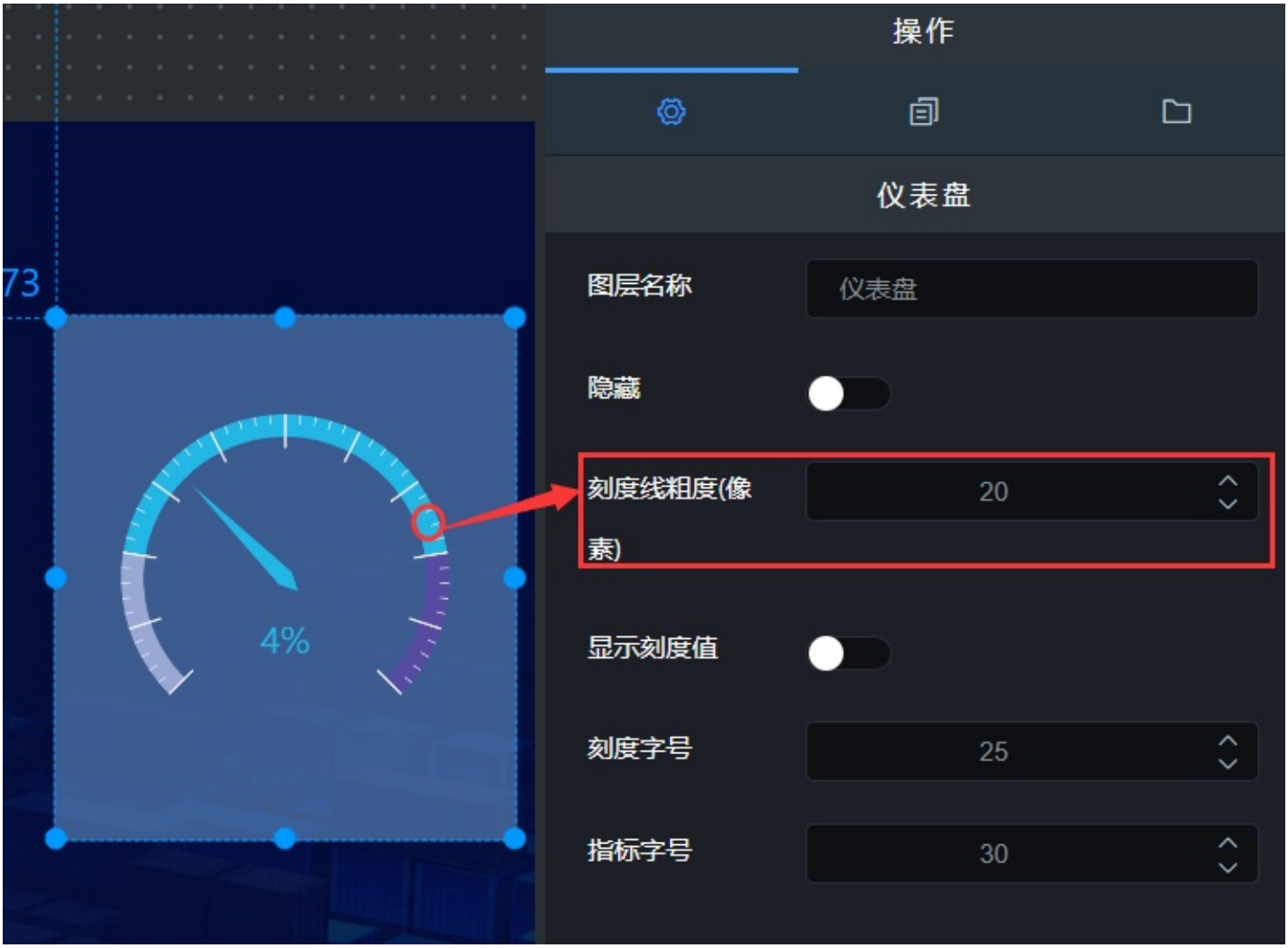


图5. 24

2. 设置刻度值

选中该仪盘表组件，在操作界面右侧，打开“显示刻度”按钮，可在仪盘表显示刻度，设置字号处可修改刻度子的大小，如图5.25。

图5. 25

3. 显示刻度线

选中该仪盘表组件，在操作界面右侧的“显示刻度线”处可设置是否显示刻度线，如图5.26。（图5.27为显示刻度线样式；图5.28为不显示刻度线样式；）

图5. 26

图 5.27　　图 5.28

3. 指标字号

选中该仪盘表组件，在操作界面右侧的“指标字号”处可设置指标名称字的大小，如图5.26。

图5. 26

三、接口设置

1. 数据类型

数据类型分为静态数据和动态数据；

- 静态数据：写死的数据；
- 动态数据：会随着接口传过来的数据实时变化；（一般这种比较常用）

2. 接口地址

（1）静态数据，接口地址穿过来的内容要符合以下格式：

```
{
    "min": 1,
    "max": 10,
    "label": "名称",
    "value": 2,
    "unit": "%"
}
```

（2）动态数据，接口地址穿过来的内容要符合以下格式：

```
{"data":{"min":1,"max":10,"label":"名称","value":4,"unit":"%"}}
```

3. 刷新时间

这个参数主要针对动态数据设置的，完成数据的实时更新。

- 如果你想设置成5秒刷新一次，可以将刷新时间设置成“5000”；

4. 刷新数据

这个参数主要是重新请求以下接口，完成数据的更新。

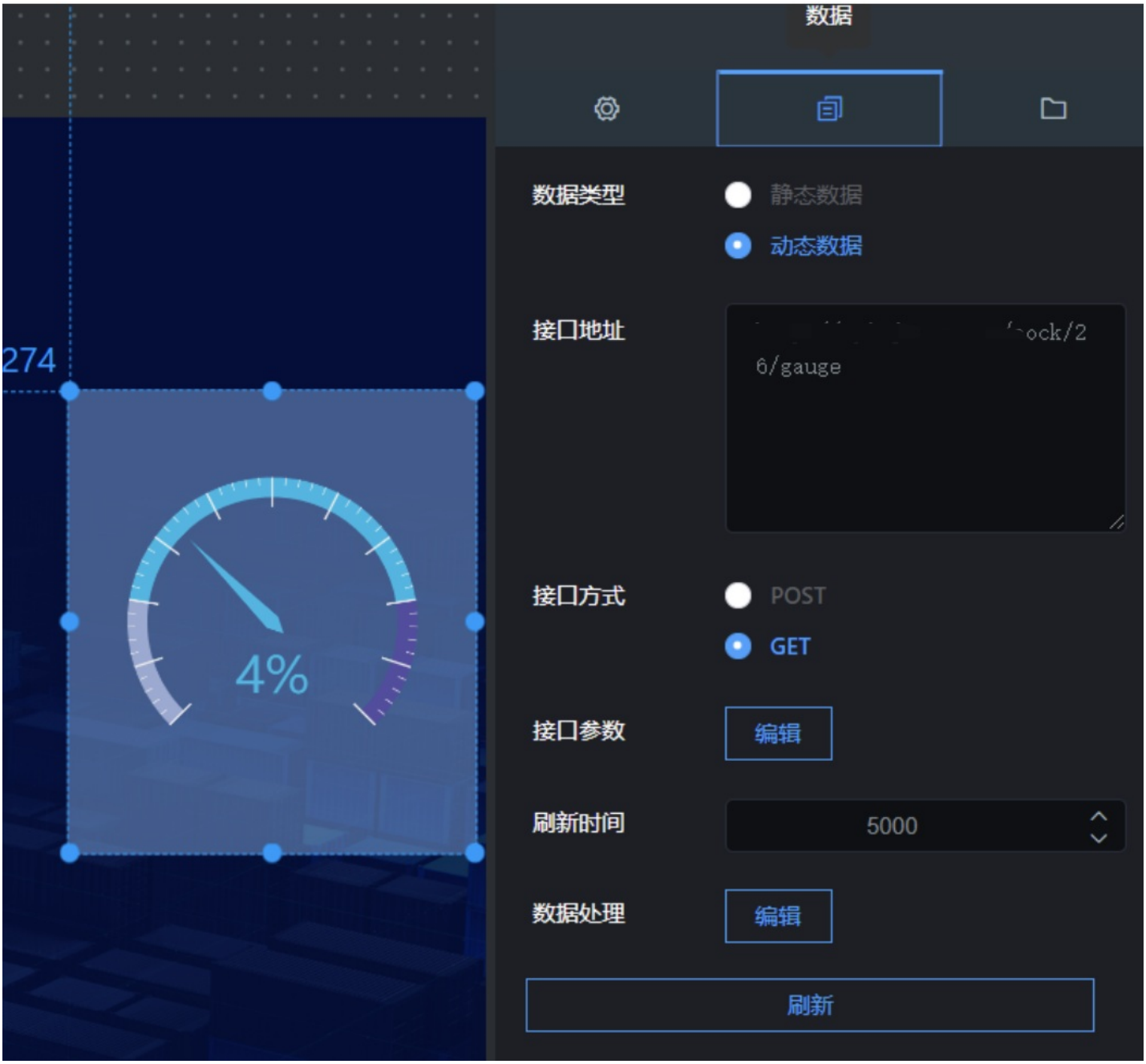

数据
数据类型
静态数据
动态数据
接口地址
ock/2
6/gauge
274
4%
接口方式
POST
GET
接口参数
编辑
刷新时间
5000
数据处理
编辑
刷新

图5.27

5.3字符云组件

字符云组件就是设置字符效果的组件，效果图如图5.31。点击“ ”图标，再点击“字符云”，即可创建字符云，如图5.32；

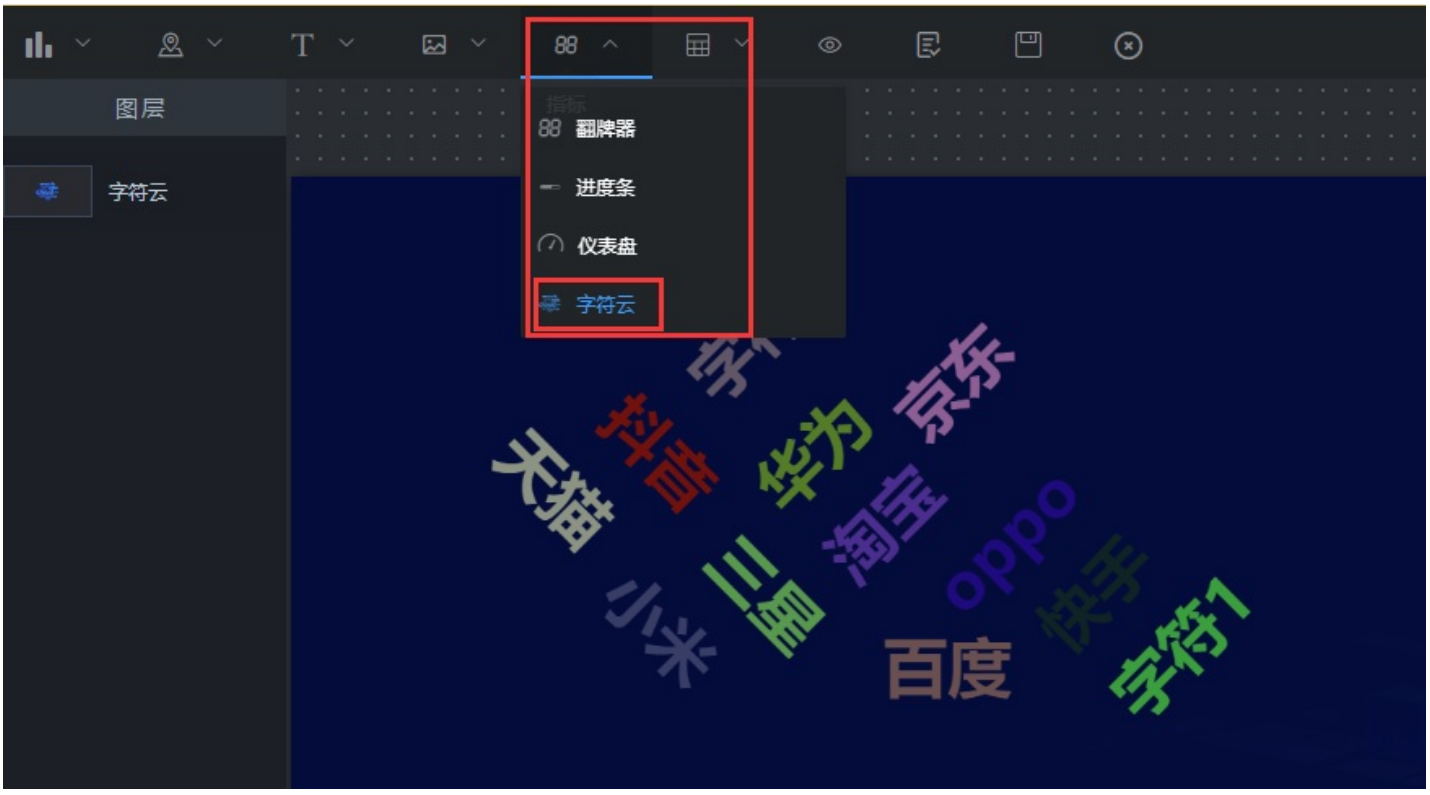


图5. 31

图5. 32

一、组件名称设置

选中该字符云组件，在操作界面右侧的“图层名称”处可修改组件的名称，如图5.33。（名称最好要设置一下，方便后期组件管理）

图5. 33

二、样式设置

1. 最小字体

选中字符云组件，在操作界面右侧，在“最小字体”设置最小字体，如图5.34；

图5.34

2. 最大字体

选中字符云组件，在操作界面右侧，在“最大字体”设置最大字体，如图5.35；

图5.35

3. 间距

选中该字符云组件，在操作界面右侧的“间距”处可设置文字的间距，如图5.36。

图5. 36

4. 旋转

- 选中该字符云组件，在操作界面右侧，打开“旋转”按钮，字符设置成旋转状态，如图5.37。
- 关闭“旋转”按钮，样式如图5.38；

图5. 37

图5. 38

三、接口设置

1. 数据类型

数据类型分为静态数据和动态数据；

- 静态数据：写死的数据；
- 动态数据：会随着接口传过来的数据实时变化；（一般这种比较常用）

2. 接口地址

（1）静态数据，接口地址穿过来的内容要符合以下格式：

```
[
    {
        "name": "三星",
        "value": "1234"
    },
    {
        "name": "小米",
        "value": "1234"
    },
    {
        "name": "华为",
        "value": "1234"
    },
    {
        "name": "oppo",
        "value": "1234"
    },
    {
        "name": "抖音",
        "value": "1234"
    },
    {
        "name": "快手",
        "value": "1234"
    },
    {
        "name": "淘宝",
        "value": "1234"
    },
    {
        "name": "百度",
        "value": "1234"
    },
    {
        "name": "京东",
        "value": "1234"
    },
    {
```

```
        "name": "天猫",
        "value": "1234"
    },
    {
        "name": "字符1",
        "value": "1234"
    },
    {
        "name": "字符1",
        "value": "1234"
    }
]
```

（2）动态数据，接口地址穿过来的内容要符合以下格式：

```
{"data":[
          {
                "name": "三星",
                "value": "1234"
            },
            {
                "name": "小米",
                "value": "1234"
            },
            {
                "name": "华为",
                "value": "1234"
            },
            {
                "name": "oppo",
                "value": "1234"
            },
            {
                "name": "抖音",
                "value": "1234"
            },
            {
                "name": "快手",
                "value": "1234"
            },
            {
                "name": "淘宝",
                "value": "1234"
            },
            {
                "name": "百度",
                "value": "1234"
            },
```

```
            {
                "name": "京东",
                "value": "1234"
            },
            {
                "name": "天猫",
                "value": "1234"
            },
            {
                "name": "字符1",
                "value": "1234"
            },
            {
                "name": "字符1",
                "value": "1234"
            }
]}
```

3. 刷新时间

这个参数主要针对动态数据设置的，完成数据的实时更新。

- 如果你想设置成5秒刷新一次，可以将刷新时间设置成“5000”；

4. 刷新数据

这个参数主要是重新请求以下接口，完成数据的更新。


操作
数据类型
静态数据
动态数据
接口地址
6/wordCloud
接口方式
POST
GET
接口参数
编辑
刷新时间
5000
数据处理
编辑
刷新
天猫
华为
三星
oppo
京东
百度
小米
快手
淘宝
字符1

图5.39

5.5进度条组件

进度条的设置请参考环形图组件。

6.表格类组件

6.1表格组件
6.2选项卡组件

6.1表格组件

表格组件就是设置表格样式的组件。点击“ ”图标，再点击“表格”，即可创建新的表格，如图6.11；

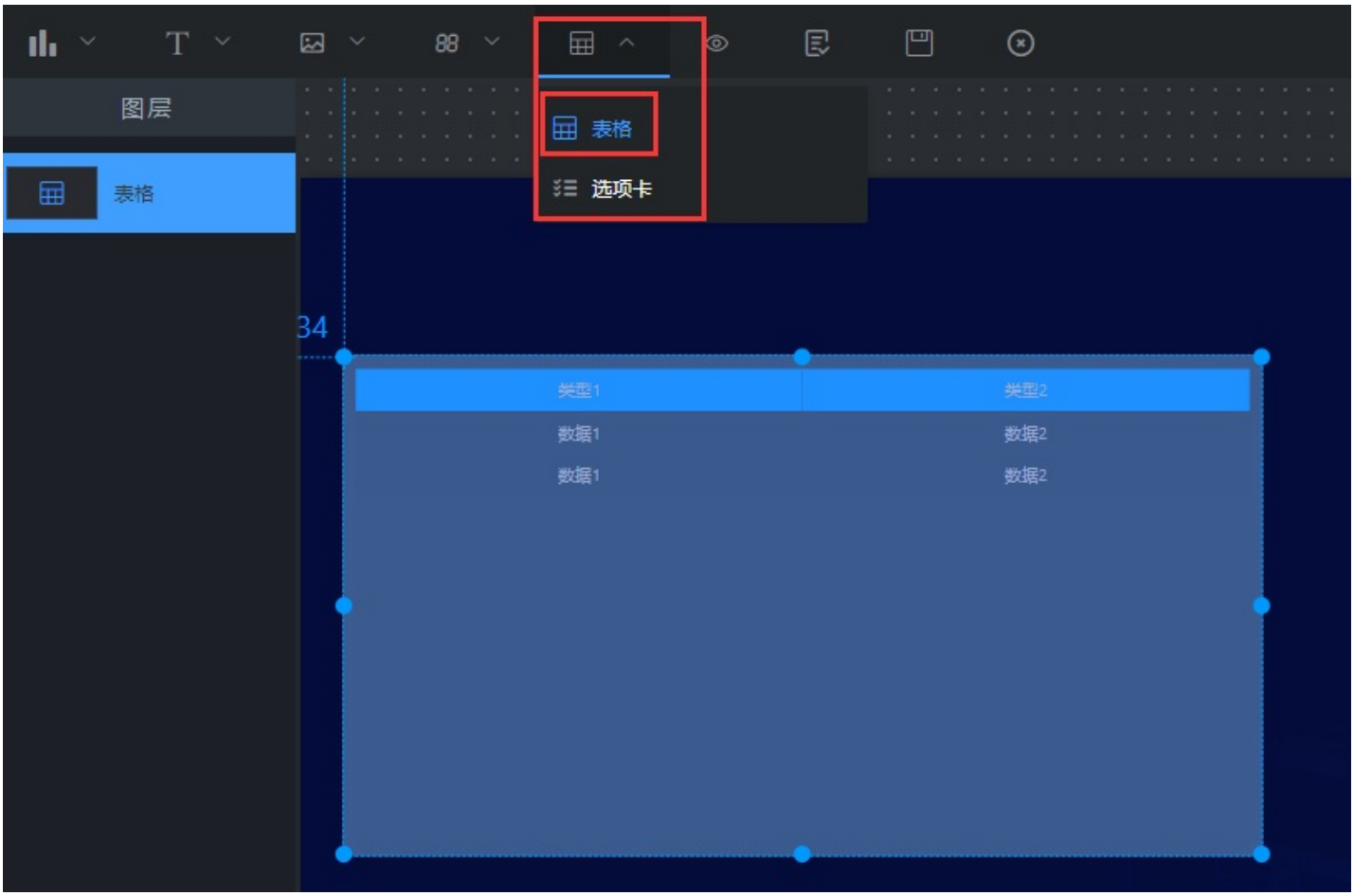


图6. 11

一、组件名称设置

选中该表格组件，在操作界面右侧的“图层名称”处可修改组件的名称，如图6.12。（名称最好要设置一下，方便后期组件管理）

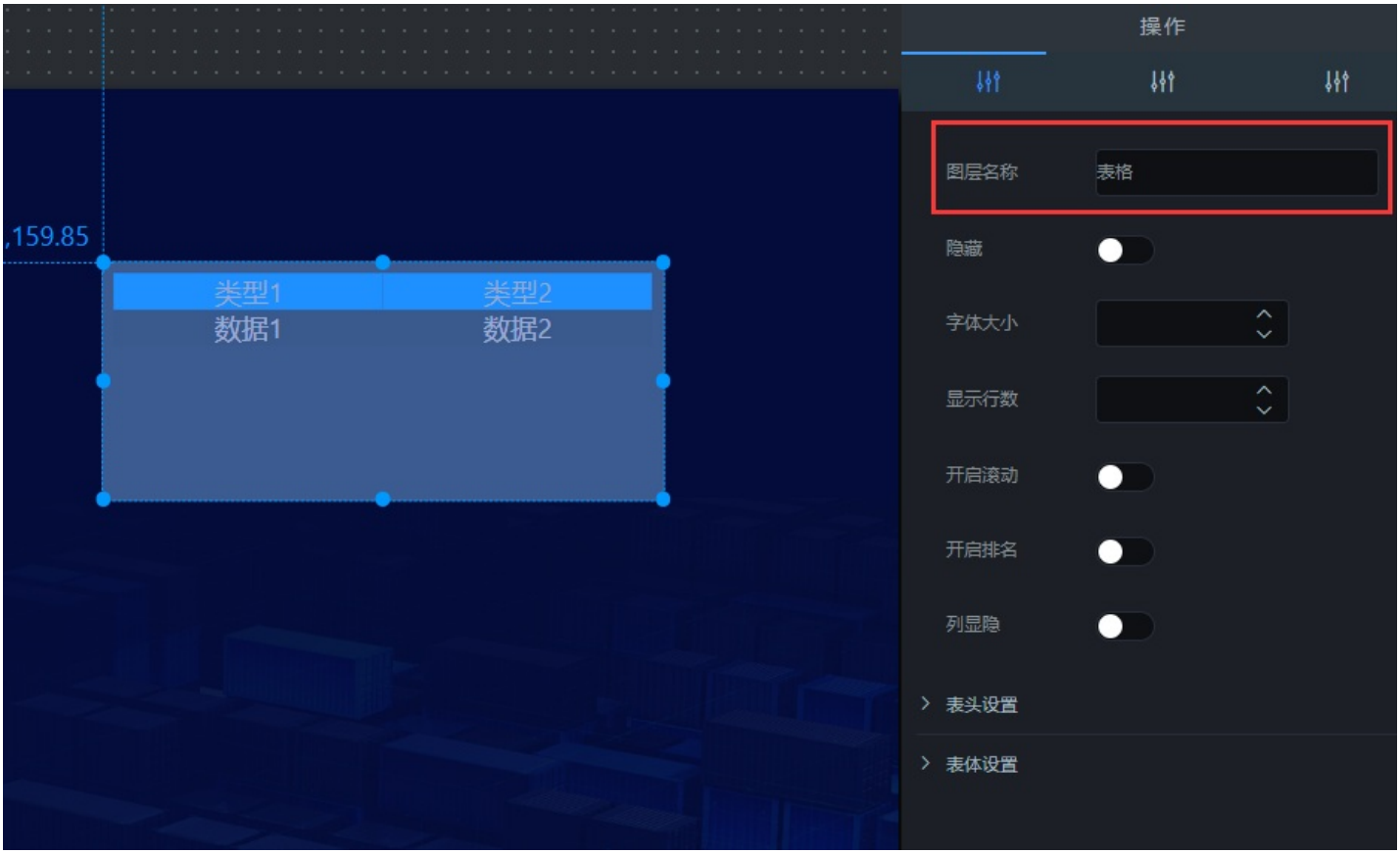


图6. 12

二、字体大小

选中该表格组件，在操作界面右侧的“字体大小”处可设置表格内文字大小，如图6.13。

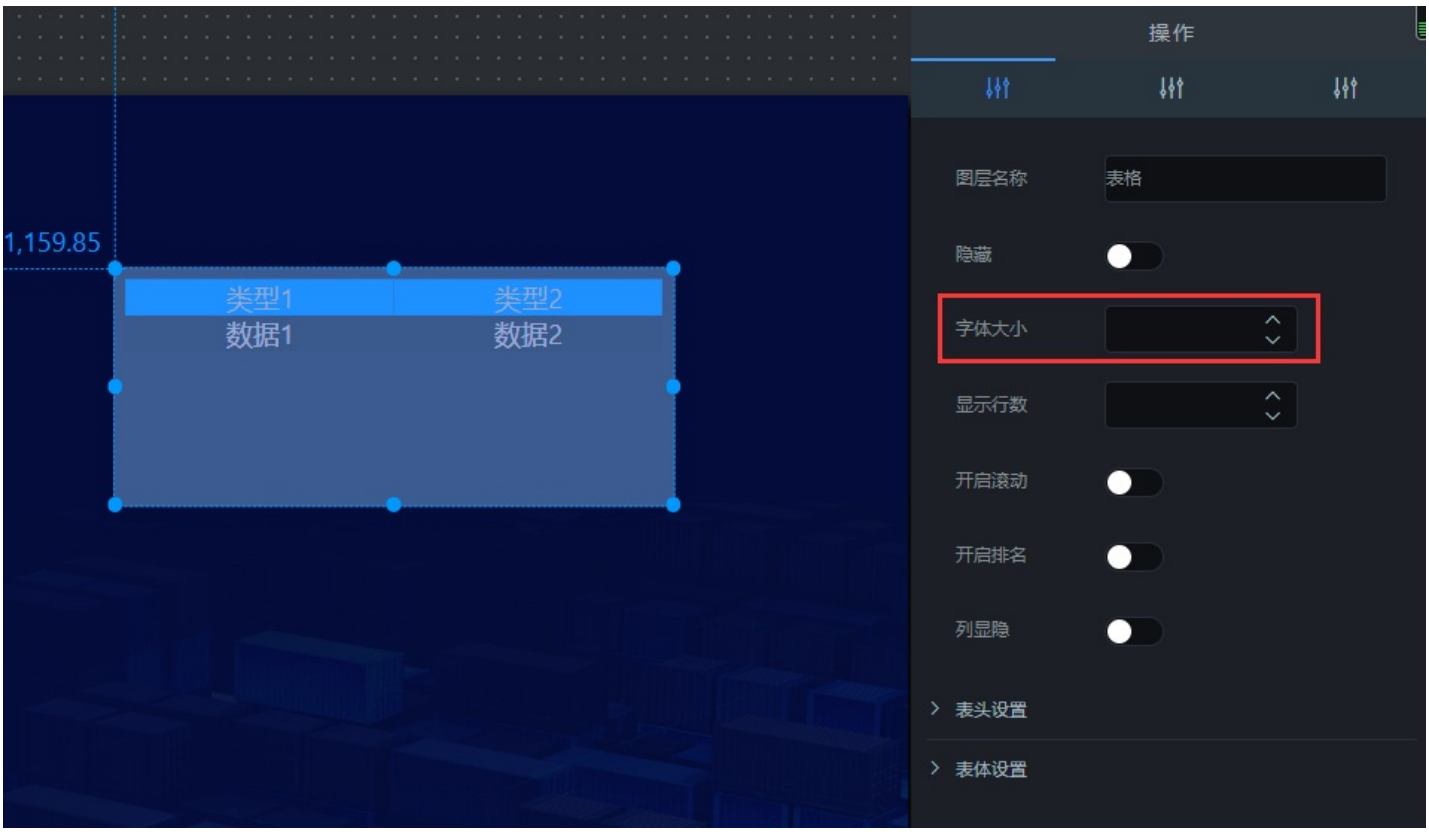


图6.13

三、显示行高

选中该表格组件，在操作界面右侧的“显示行高”处可设置表格的行高，如图6.14。

备注：从1开始，数字越大，行高越小

图6.14

四、开启滚动

选中该表格组件，在操作界面右侧，打开“开启滚动”开关，输入滚动时间、滚动行数，设置表格的滚动样式，如图6.15。

- 滚动时间：多久滚动一次；（单位毫秒）
- 滚动行数：每滚动一次，滚动几行；

图6.15

五、开启排名

开启排名之后，会在页面显示排名样式，如图6.16；

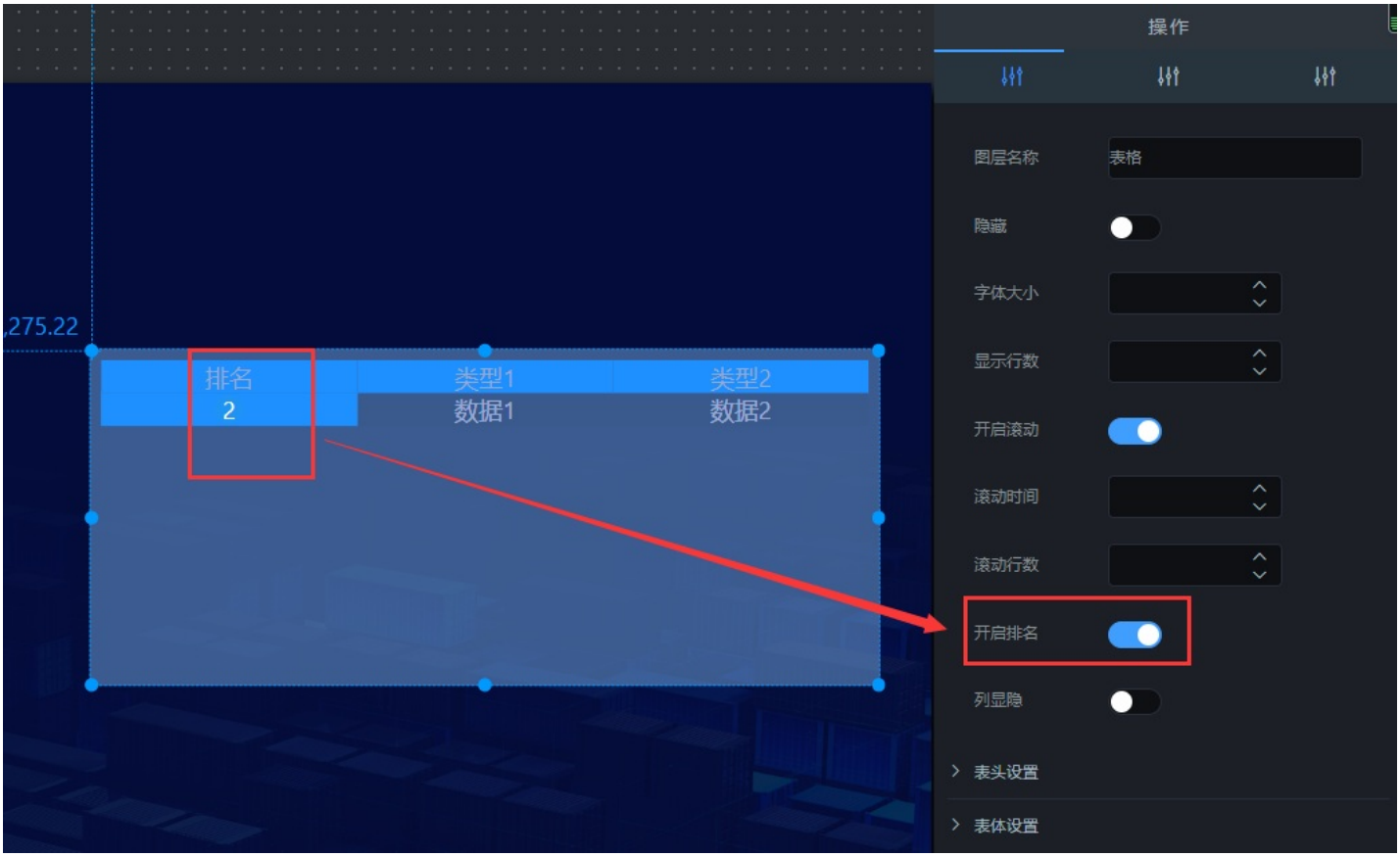


图6.16

六、列显隐

选中该表格组件，在操作界面右侧，打开“开启显隐”开关，可在页面设置显示和隐藏某一列，如图6.17。

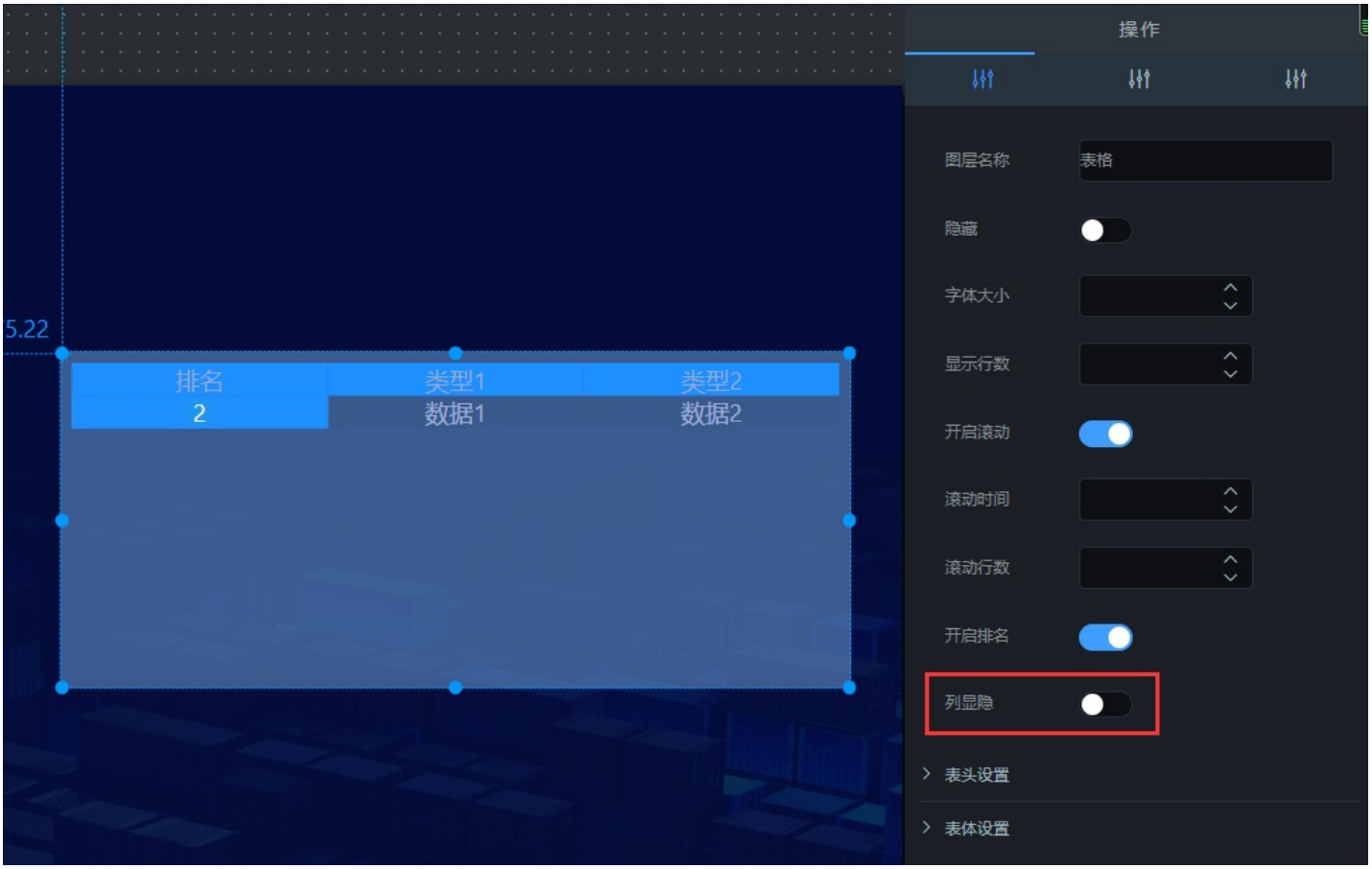


图6. 17

七、表头设置

选中该表格组件，在操作界面右侧的“表头设置”处可设置表头的样式，如图6.18。

- 表头显隐：设置表头是否显示；
- 表头颜色：设置表头文字颜色；
- 表头背景：设置表头的背景颜色；

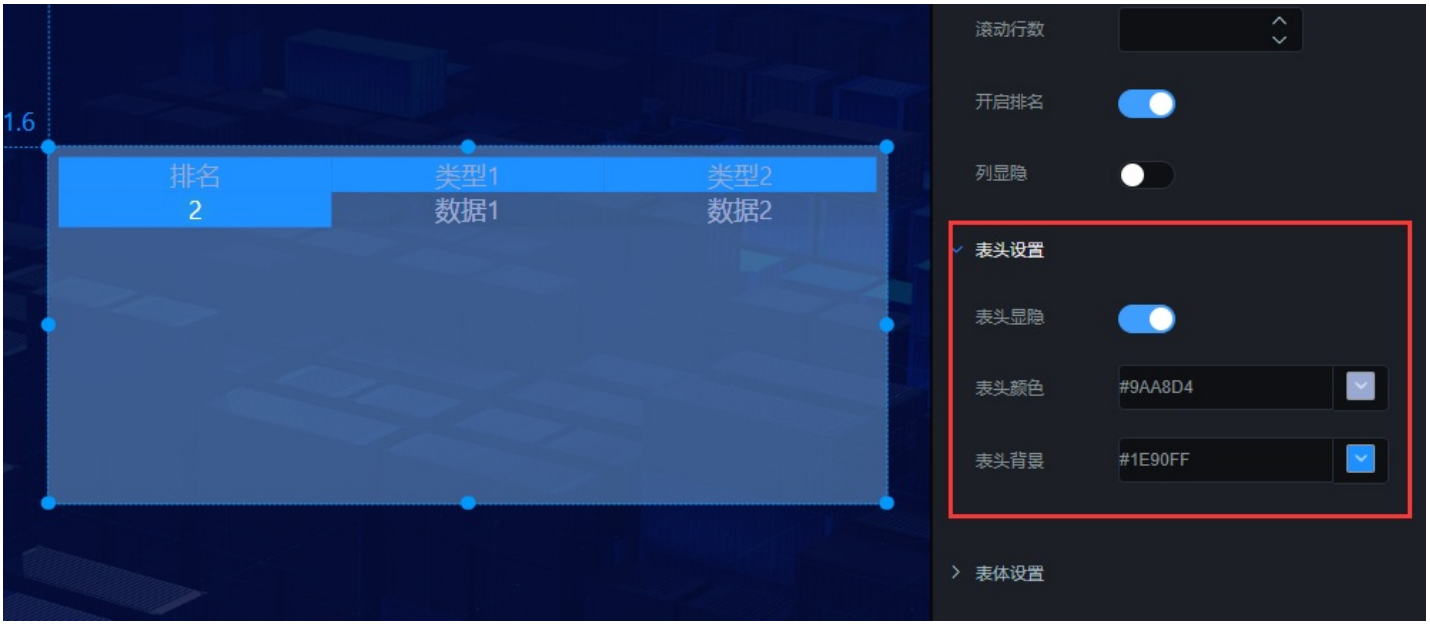


图6. 18

八、表格设置

选中该表格组件，在操作界面右侧的“表格设置”处可设置表格的样式，如图6.19。

- 文字颜色：设置表格文字颜色；
- 表格背景：设置表格的背景颜色；
- 边框宽度：设置表格边框的粗细大小；
- 边框颜色：设置表格变宽的颜色；
- 边框样式：包含实线、虚线和隐藏；
- 奇行颜色：设置表头奇数行的颜色；
- 偶行颜色：设置表格偶数行的颜色；

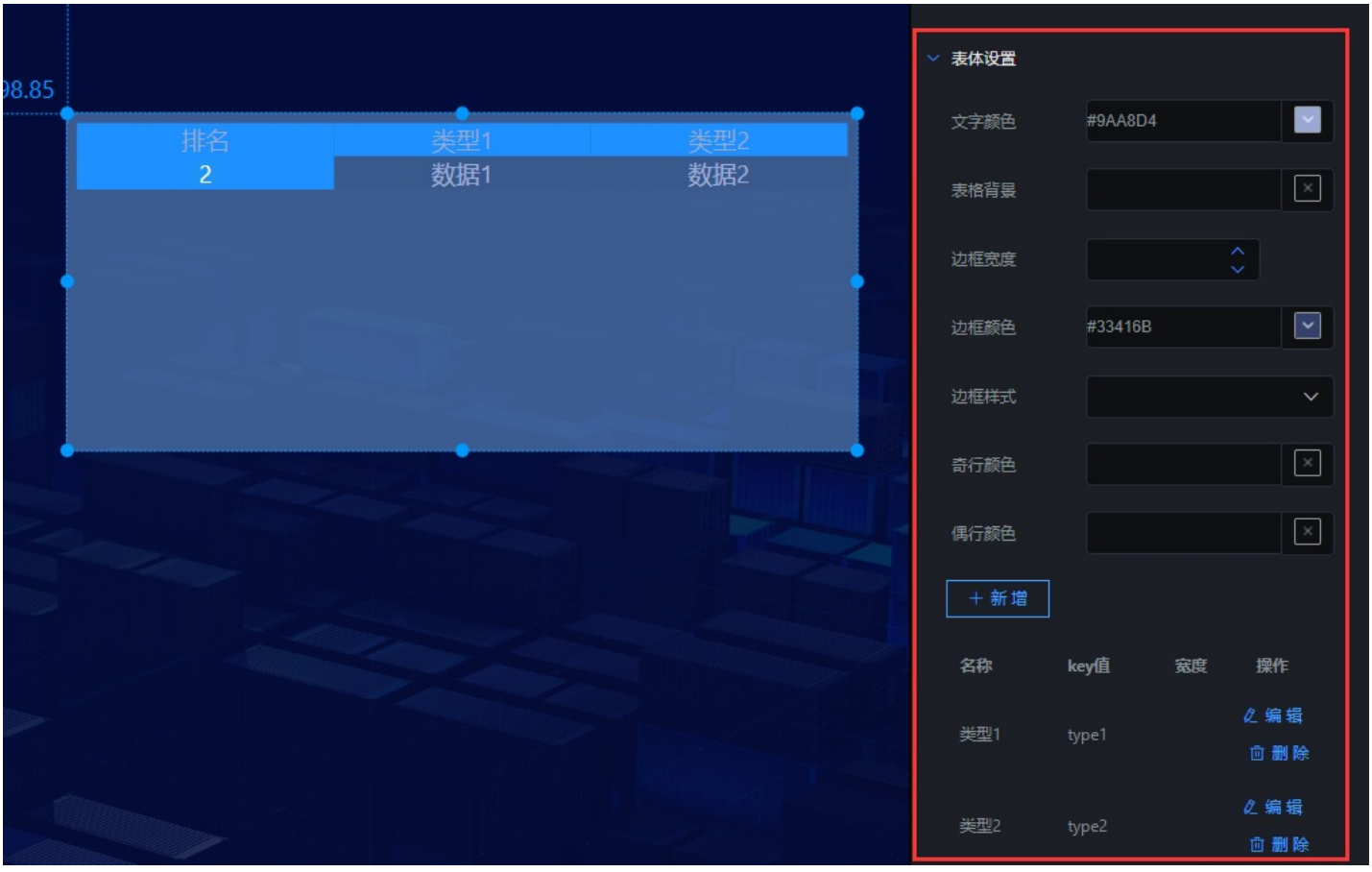


图6. 19

九、表头数据设置

选中该表格组件，在操作界面右侧的最底下，可设置表头名称，key值要跟接口中的想对应。

备注：如果接口传过来的数据是3列，你需添加3行数据，并且key值要跟接口中参数想对应，如图6.191、6.192；

图6.191

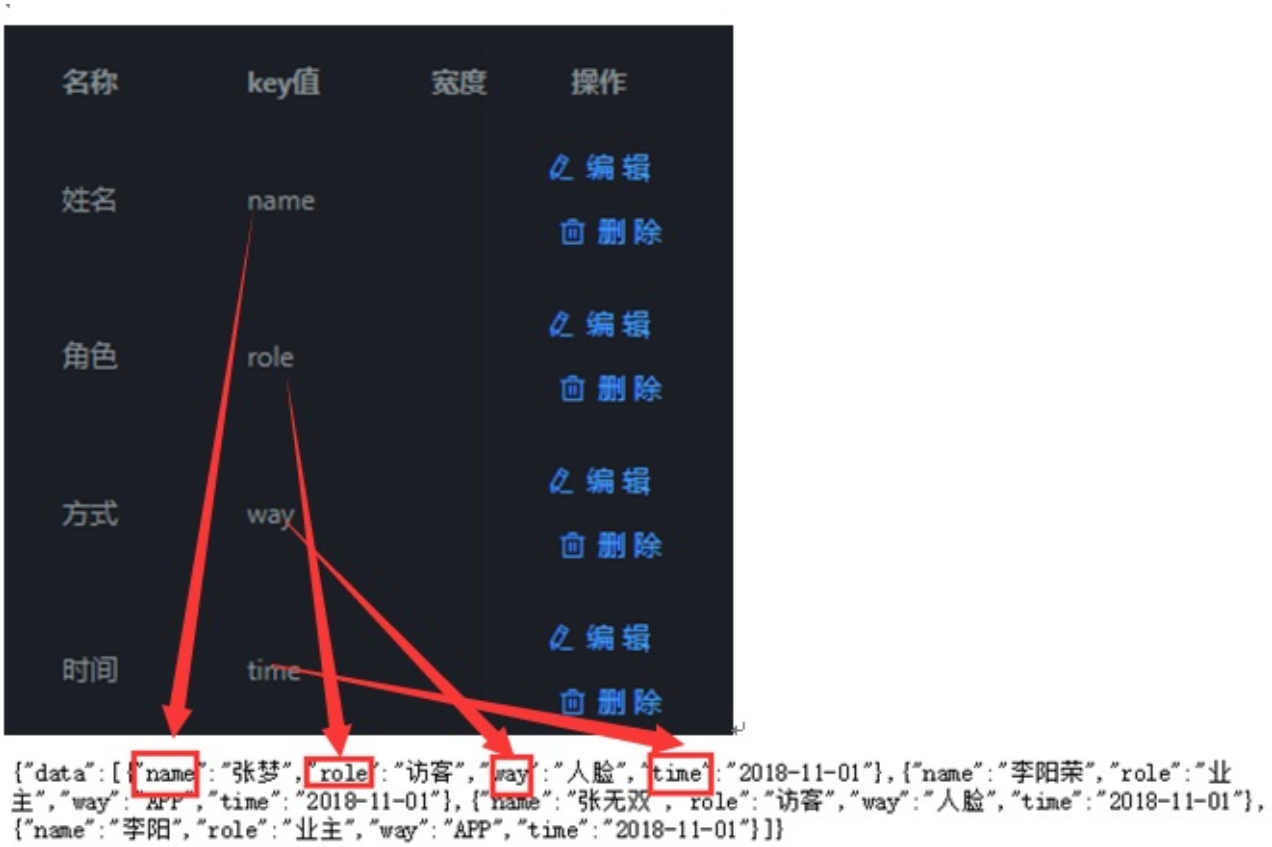


图6.192

十、接口设置

选中该表格组件，在操作界面右侧，点击“  ”，可设置接口，如图6.193。

1. 数据类型

数据类型分为静态数据和动态数据；

- 静态数据：写死的数据；
- 动态数据：会随着接口传过来的数据实时变化；（一般这种比较常用）

2. 接口地址

（1）静态数据，接口地址传过来的内容要符合以下格式：

```
[{"name":"张梦","role":"访客","way":"人脸","time":"2018-11-01"},{"name":"李阳荣","role":"业主","way":"APP","time":"2018-11-01"},{"name":"张无双","role":"访客","way":"人脸","time":"2018-11-01"},{"name":"李阳","role":"业主","way":"APP","time":"2018-11-01"}]
```

（2）动态数据，接口地址传过来的内容要符合以下格式：

```
{"data":[{"name":"张梦","role":"访客","way":"人脸","time":"2018-11-01"},{"name":"李阳荣","role":"业主","way":"APP","time":"2018-11-01"},{"name":"张无双","role":"访客","way":"人脸","time":"2018-11-01"},{"name":"李阳","role":"业主","way":"APP","time":"2018-11-01"}]}
```

3. 刷新时间

这个参数主要针对动态数据设置的，完成数据的实时更新。

- 如果你想设置成5秒刷新一次，可以将刷新时间设置成“5000”；

4. 刷新数据

这个参数主要是重新请求以下接口，完成数据的更新。


数据类型
静态数据
动态数据
接口地址
moc
k/26/bar
接口方式
POST
GET
刷新时间
5000
数据处理

图6.193

6.2选项卡组件

选项卡组件就是设置选项卡样式的组件。点击“ ”图标，再点击“选项卡”，即可创建新的表格，如图6.21；

图6. 21

一、组件名称设置

选中该选项卡组件，在操作界面右侧的“图层名称”处可修改组件的名称，如图6.22。（名称最好要设置一下，方便后期组件管理）

图6. 22

二、字体设置

选中该选项卡组件，在操作界面右侧的“字体大小”、“字体颜色”和“字体间距”处可修改组件的字体样式，如图6.23。

- 字体大小：可修改文字的大小；
- 字体颜色：可修改文字的颜色；
- 字体间距：可修改“选项卡1”与“选项卡2”之间的距离；

图6. 23

三、边框设置

选中该选项卡组件，在操作界面右侧的“边框设置”处可设置选项卡的边框样式，如图6.24。

- 背景颜色：选项卡的背景颜色；
- 缩略图：背景图缩略图；
- 背景图片：背景图片上传；
- 边框颜色：边框的颜色；
- 边框宽度：边框宽度；

图6. 24

四、高亮设置

选中该选项卡组件，在操作界面右侧的“高亮设置”处可设置选项卡的选中的样式，如图6.25。

- 字体高亮颜色：设置选中的字体颜色；
- 缩略图：背景图缩略图；
- 背景图片：背景图片上传；
- 边框颜色：边框的颜色；
- 边框宽度：边框宽度；

图6. 25

五、选项卡切换

- 要想实现选项卡动态切换，需要在“”中选择“子类”，填写参数，就可实现选择子类数据的动态切换；
- 子类：想要实现通过选项卡切换，实现数据切换的组件，可以是柱形图，也可以是环形图等；
- 参数：随便写，可以是key;
- 本组件只可实现一个组件的数据动态切换，不能实现切换的时候，组件样式变化；

图6. 26

五、接口设置

选中该柱形图组件，在操作界面右侧，点击“  ”，可设置接口，如图6.27。

1. 数据类型

数据类型分为静态数据和动态数据；

- 静态数据：写死的数据；
- 动态数据：会随着接口传过来的数据实时变化；（一般这种比较常用）

2. 接口地址

（1）静态数据，接口地址传过来的内容要符合以下格式：

```
[
    {
        "label": "选项卡1",
        "value": "1"
    },
    {
        "label": "选项卡2",
        "value": "2"
    }
]
```

（2）动态数据，接口地址传过来的内容要符合以下格式：

```
{"data":[{"label":"选项卡1","value":"1"},{"label":"选项卡2","value":"2"},{"label"
:"选项卡3","value":"3"}]}
```

（3）如果有多个选项，就在接口中添加多个选项就可以。比如有3个选项，接口格式如下：

```
[
    {
        "label": "选项卡1",
        "value": "1"
    },
    {
        "label": "选项卡2",
        "value": "2"
    }
```

```
,
   {
        "label": "选项卡3",
        "value": "3"
    }
]
```

3. 刷新时间

这个参数主要针对动态数据设置的，完成数据的实时更新。

- 如果你想设置成5秒刷新一次，可以将刷新时间设置成“5000”；

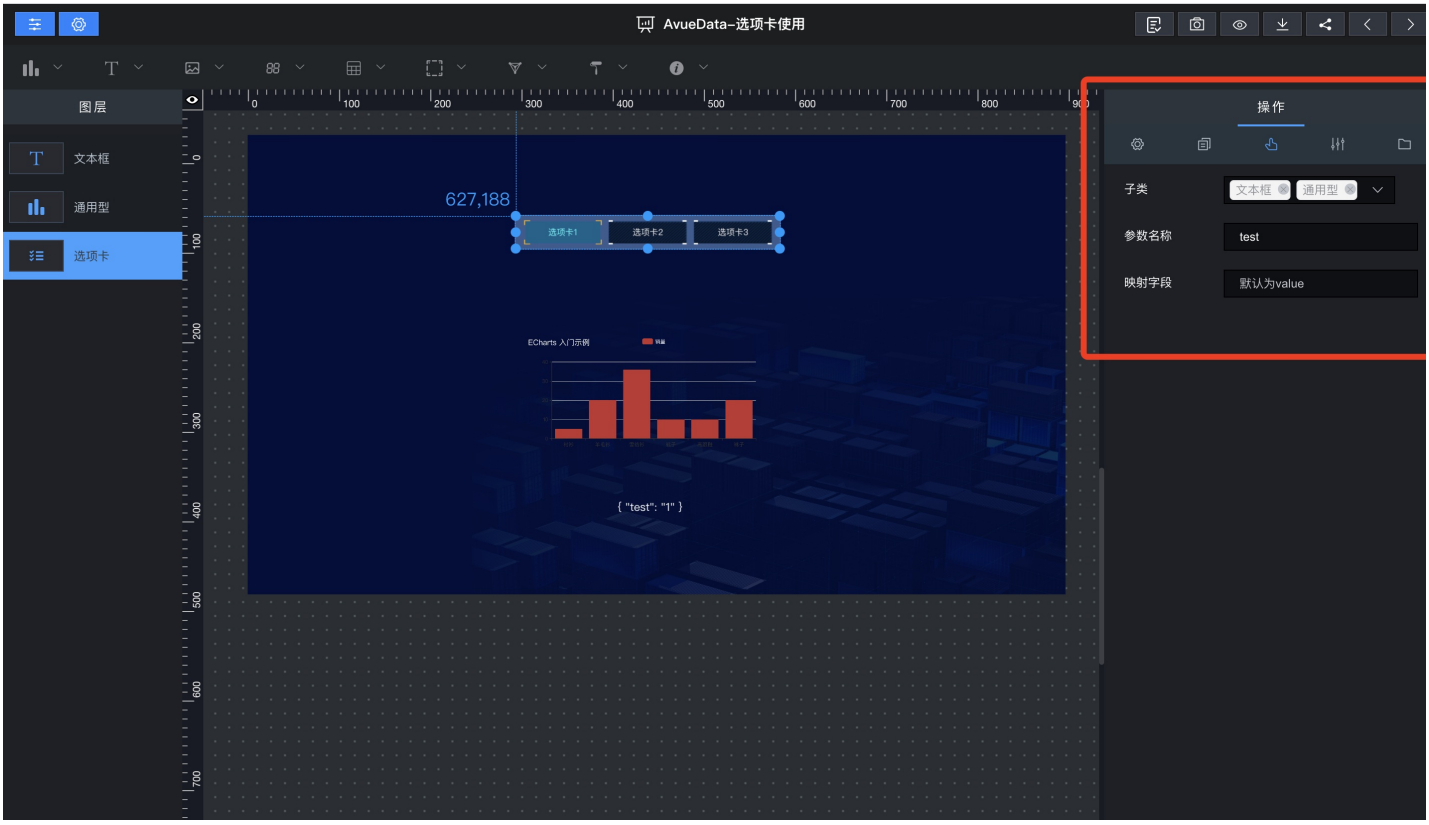
4. 刷新数据

这个参数主要是重新请求以下接口，完成数据的更新。

图6.27

六、选项卡交互

点击我跳转参考例子



选择需要交互的组件，设置传递给组件的对应参数，他会自动加到请求参数里

它默认取的是数据中的value，也可以设置取其他字段

7地图类组件

7.1地图组件

7.1地图组件

地图组件就是添加地图样式的组件。点击“  ”图标，再点击“地图”，即可创建新的图像，如图7.11；

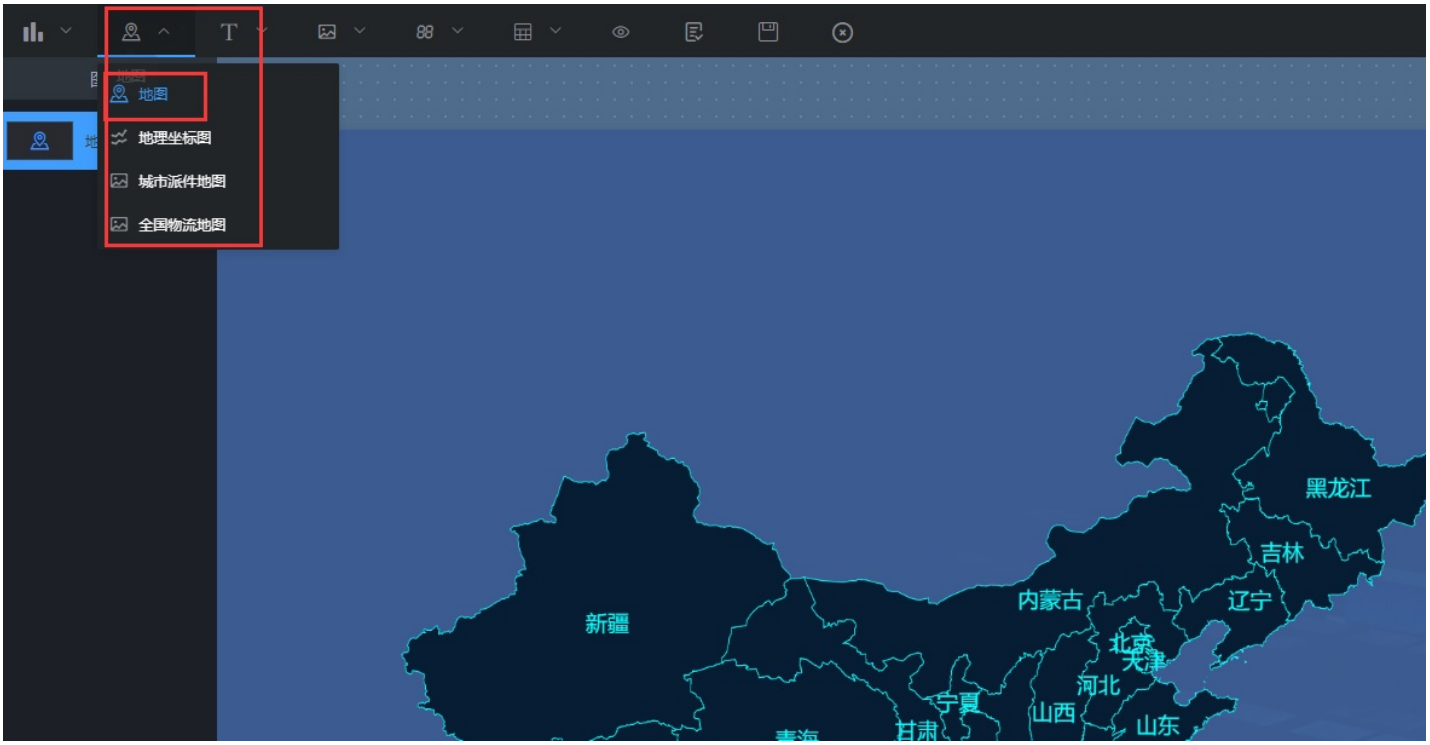


图7.11

一、组件名称设置

选中地图组件，在操作界面右侧的“图层名称”处可修改组件的名称，如图7.12。（名称最好要设置一下，方便后期组件管理）

图7. 12

二、轮播

选中该地图组件，在操作界面右侧，打开“开启轮播”开关，在“轮播时间”文本框中输入时间，来设置地图组件的轮播样式，如图7.13。

- 如果轮播时间想设为5秒，可在轮播时间的文本框中填写5000；

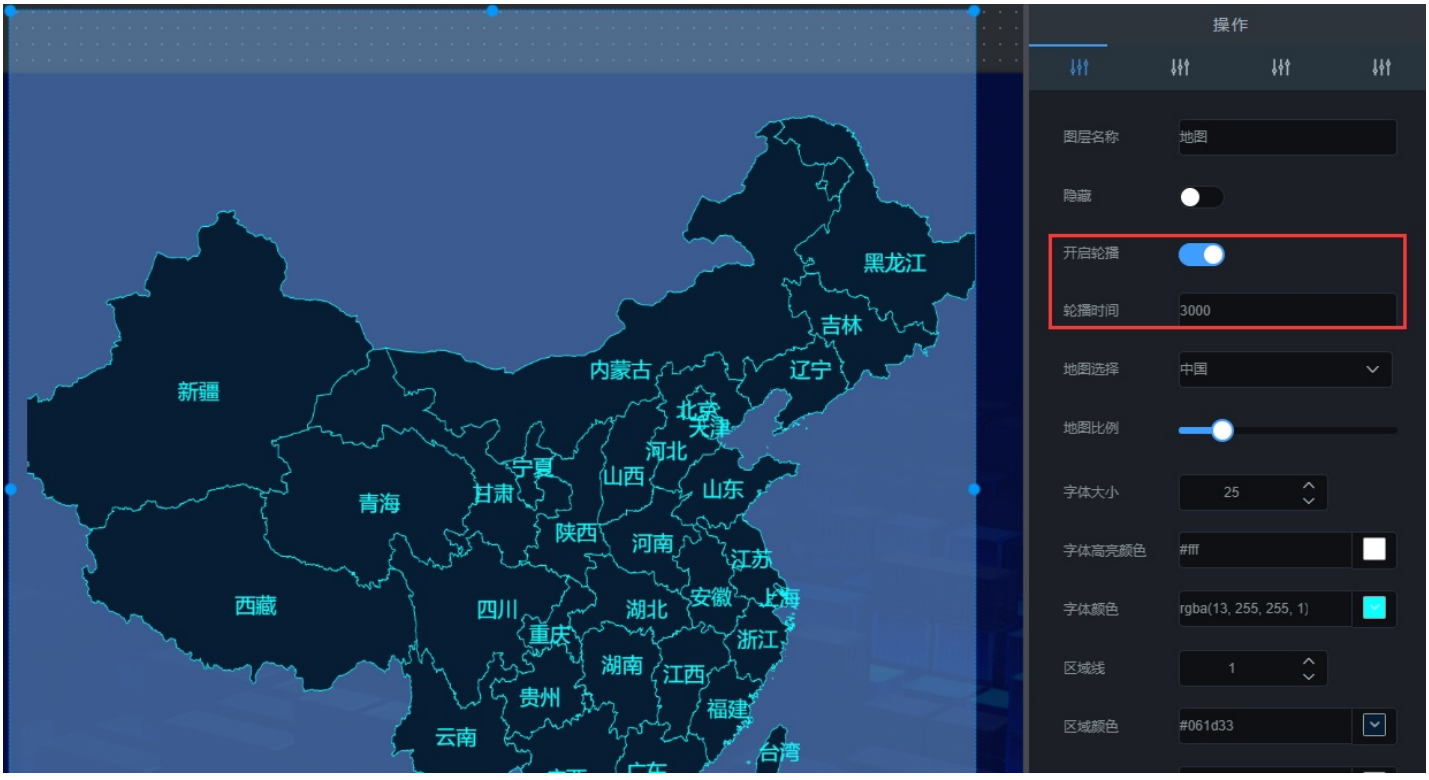


图7.13

三、地图选择

选中该地图组件，在操作界面右侧的“地图选择”处通过选择不同的地区来改变地图样式，如图7.14。

图7.14

四、地图比例

选中该地图组件，在操作界面右侧，拖动“地图比例”大小，来设置地图样式，如图7.15。

图7.15

五、字体设置

选中该地图，在操作界面右侧的“字体设置”处可修改地图组件的文字样式，如图7.16。

- 字体大小：文字的大小；
- 字体高亮颜色：鼠标点击的时候高亮显示的颜色；
- 字体颜色：文字的颜色；

图7.16

六、区域设置

选中该地图组件，在操作界面右侧的“提示语设置”处可修改地图组件的提示语，如图7.17。

- 区域线：提示语的字体大小；
- 区域颜色：提示语的字体颜色；
- 区域高亮颜色：提示语的字体颜色；

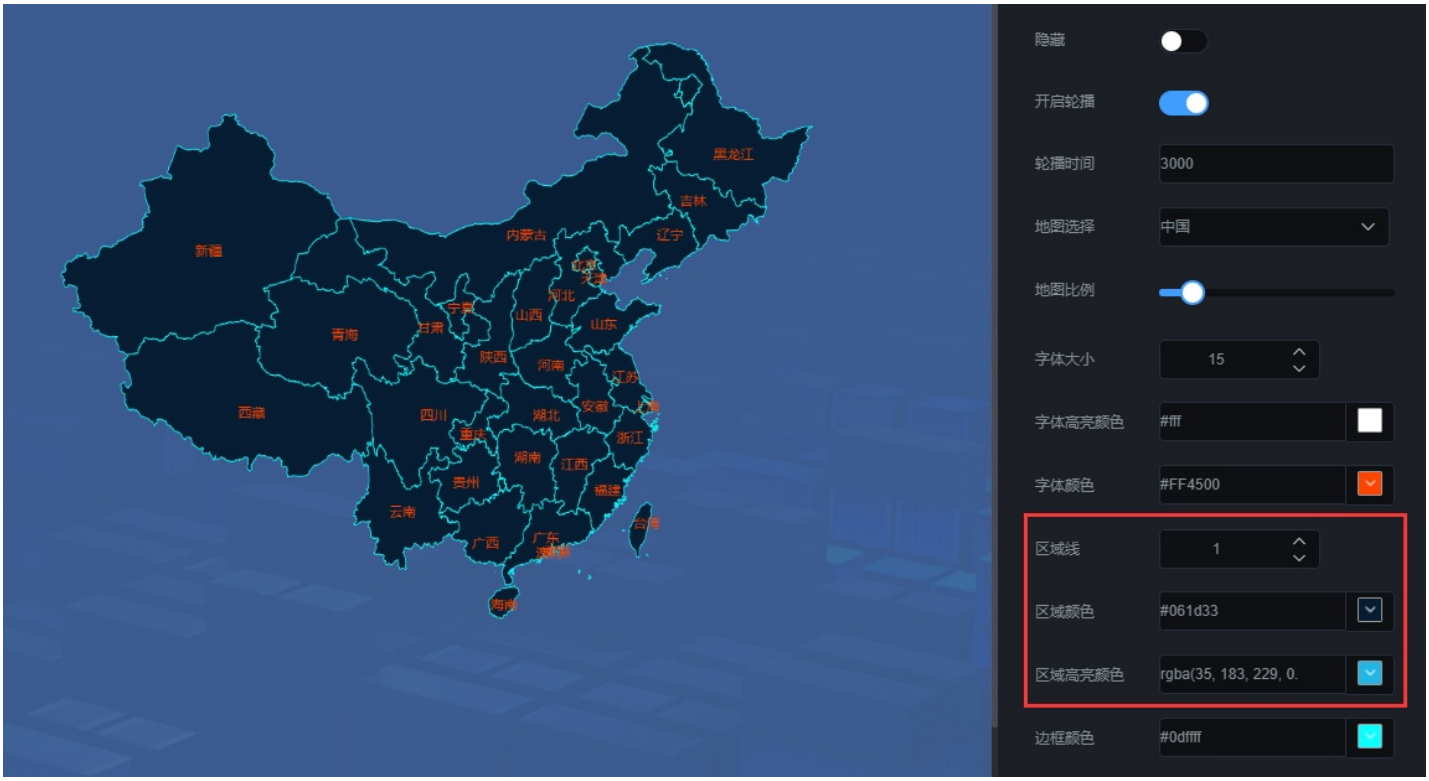


图7. 17

七、边框颜色

选中该地图组件，在操作界面右侧，通过设置边框颜色，来设置地图上区域之间的颜色样式，如图7.18。

图7. 18

八、接口设置

选中该漏斗图组件，在操作界面右侧，点击“

”，可设置接口，如图7.19。

1. 数据类型

数据类型分为静态数据和动态数据；

- 静态数据：写死的数据；
- 动态数据：会随着接口传过来的数据实时变化；

2. 接口地址

（1）静态数据，接口地址传过来的内容要类似以下格式：

```
[{"name":"测试坐标1","value":1,"lng":118.30078125,"lat":36.91915611148194,"zoom"
:1},{"name":"测试坐标2","value":1,"lng":112.21435546875,"lat":37.965854128749434
,"zoom":1}]
```

（2）动态数据，接口地址传过来的内容要类似以下格式：

```
{"data":[{"name":"测试坐标1","value":1,"lng":118.30078125,"lat":36.9191561114819
4,"zoom":1},{"name":"测试坐标2","value":1,"lng":112.21435546875,"lat":37.9658541
28749434,"zoom":1}]}
```

3. 刷新时间

这个参数主要针对动态数据设置的，完成数据的实时更新。

- 如果你想设置成5秒刷新一次，可以将刷新时间设置成“5000”；

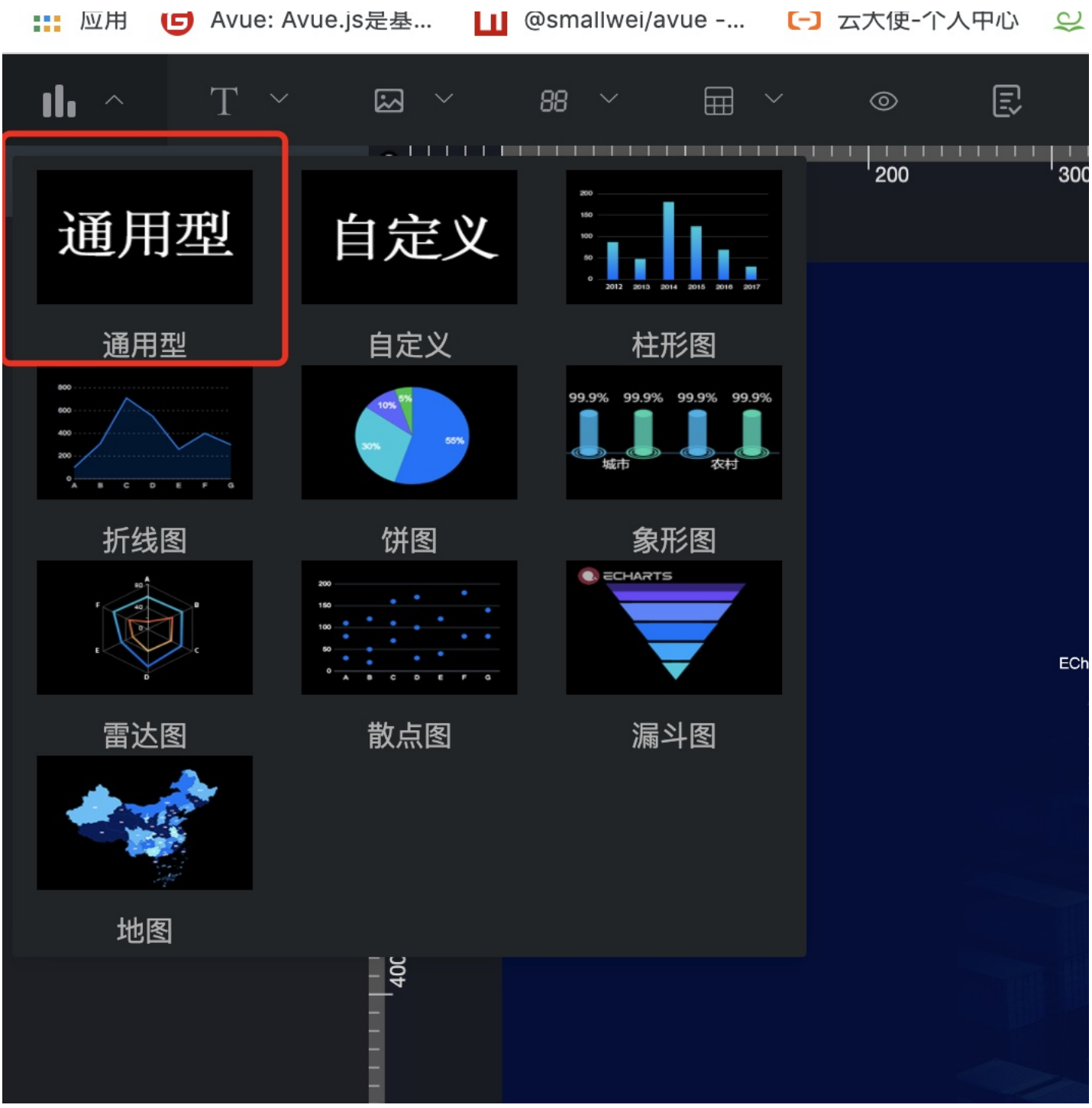
4. 刷新数据

这个参数主要是重新请求以下接口，完成数据的更新。


图层名称
城市派件地图
隐藏
数据类型
静态数据
动态数据
地图选择
中国
接口地址
接口方式
POST
GET
刷新时间
10000

图7.19

8万能组件

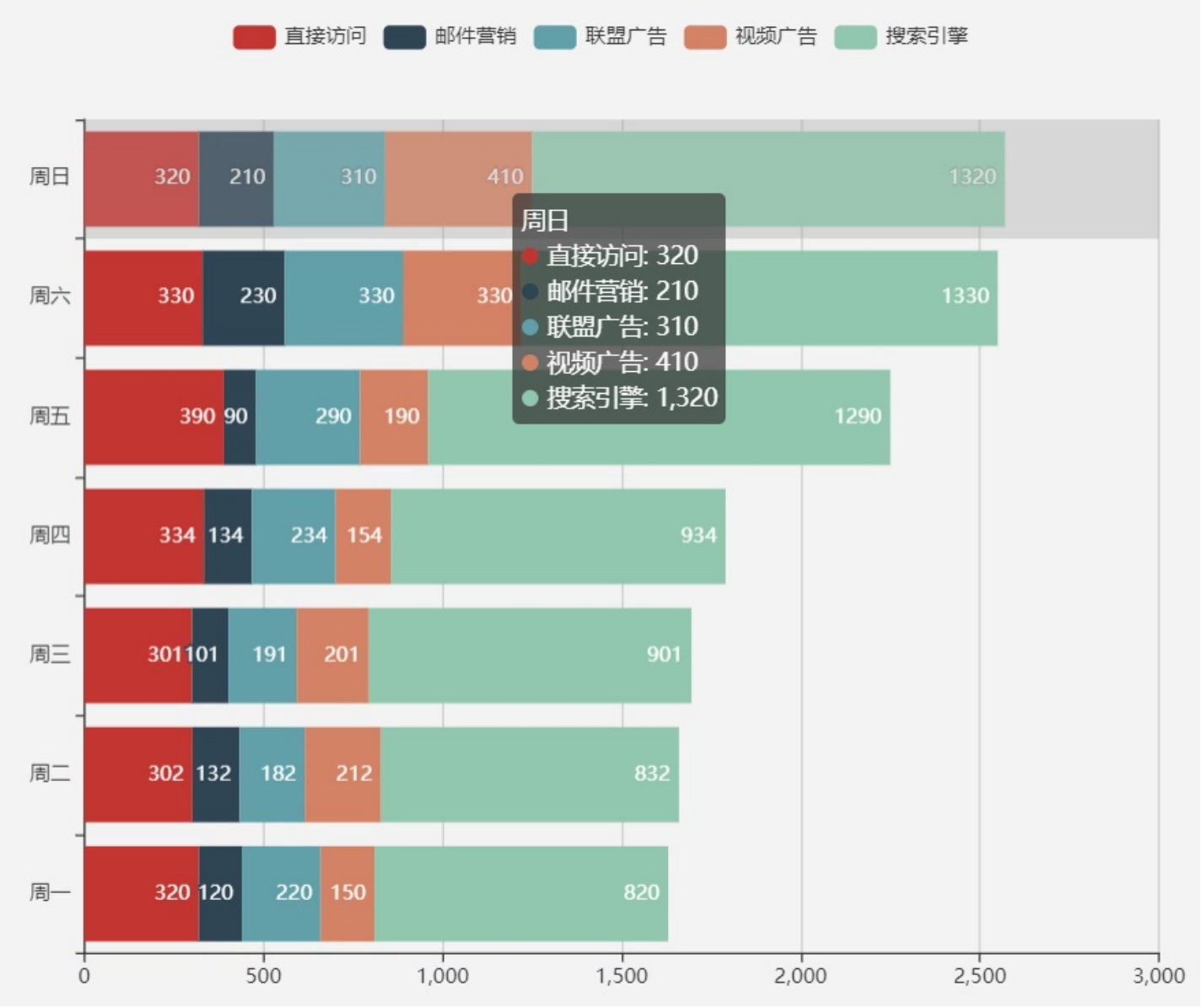

应用
Avue: Avue.js是基...
@smallwei/avue -...
云大使-个人中心
通用型
自定义
柱形图
折线图
饼图
象形图
雷达图
散点图
漏斗图
地图
ECHARTS
99.9%
城市
农村

参考了例子

8.1堆叠条形图

通过万能组件配置堆叠条形图

案例效果



代码实现

```
option = {
    tooltip: {
        trigger: 'axis',
        axisPointer: {            // 坐标轴指示器，坐标轴触发有效
            type: 'shadow'        // 默认为直线，可选为：'line' | 'shadow'
        }
    },
    legend: {
        data: ['直接访问', '邮件营销', '联盟广告', '视频广告', '搜索引擎']
    },
    grid: {
```

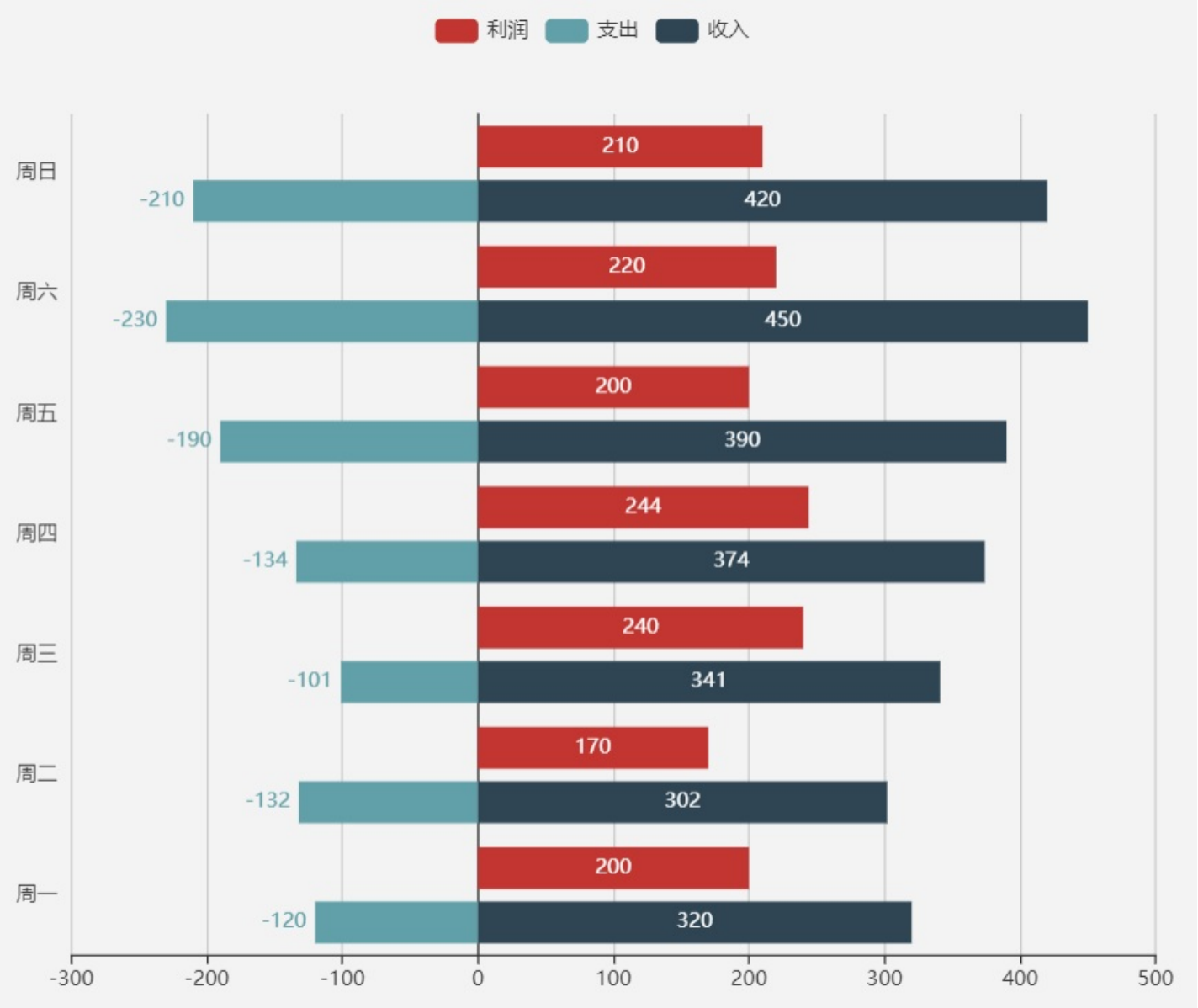
```
        left: '3%',
        right: '4%',
        bottom: '3%',
        containLabel: true
    },
    xAxis: {
        type: 'value'
    },
    yAxis: {
        type: 'category',
        data: ['周一', '周二', '周三', '周四', '周五', '周六', '周日']
    },
    series: [
        {
            name: '直接访问',
            type: 'bar',
            stack: '总量',
            label: {
                show: true,
                position: 'insideRight'
            },
            data: [320, 302, 301, 334, 390, 330, 320]
        },
        {
            name: '邮件营销',
            type: 'bar',
            stack: '总量',
            label: {
                show: true,
                position: 'insideRight'
            },
            data: [120, 132, 101, 134, 90, 230, 210]
        },
        {
            name: '联盟广告',
            type: 'bar',
            stack: '总量',
            label: {
                show: true,
                position: 'insideRight'
            },
            data: [220, 182, 191, 234, 290, 330, 310]
        },
        {
            name: '视频广告',
            type: 'bar',
            stack: '总量',
            label: {
                show: true,
                position: 'insideRight'
```

```
            },
            data: [150, 212, 201, 154, 190, 330, 410]
        },
        {
            name: '搜索引擎',
            type: 'bar',
            stack: '总量',
            label: {
                show: true,
                position: 'insideRight'
            },
            data: [820, 832, 901, 934, 1290, 1330, 1320]
        }
    ]
};
```

8.2正负条形图

通过万能组件配置正负条形图

案例效果



代码实现

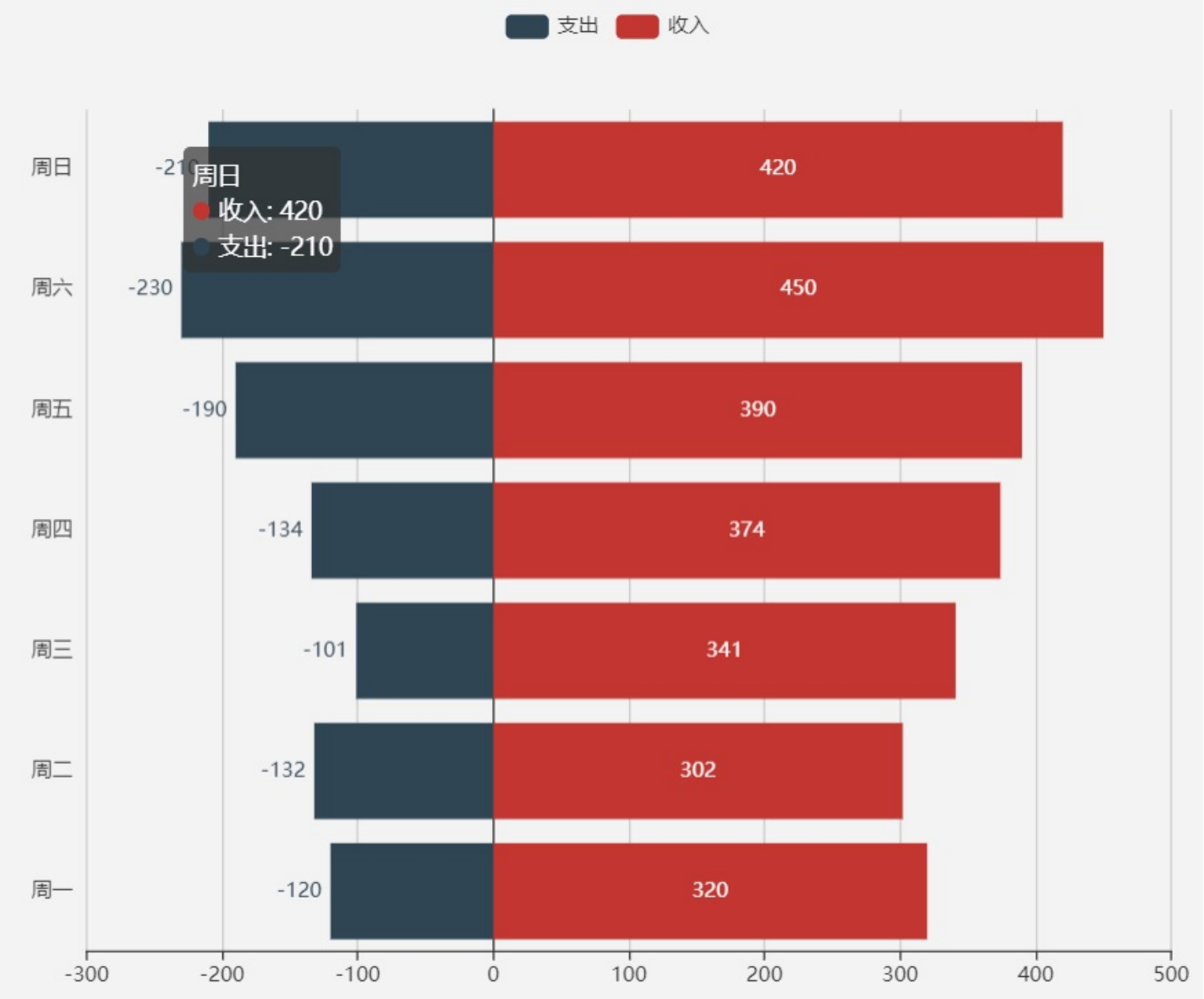
```
option = {
    tooltip: {
        trigger: 'axis',
        axisPointer: {            // 坐标轴指示器，坐标轴触发有效
            type: 'shadow'        // 默认为直线，可选为：'line' | 'shadow'
        }
    },
    legend: {
        data: ['利润', '支出', '收入']
    },
    grid: {
```

```
        left: '3%',
        right: '4%',
        bottom: '3%',
        containLabel: true
    },
    xAxis: [
        {
            type: 'value'
        }
    ],
    yAxis: [
        {
            type: 'category',
            axisTick: {
                show: false
            },
            data: ['周一', '周二', '周三', '周四', '周五', '周六', '周日']
        }
    ],
    series: [
        {
            name: '利润',
            type: 'bar',
            label: {
                show: true,
                position: 'inside'
            },
            data: [200, 170, 240, 244, 200, 220, 210]
        },
        {
            name: '收入',
            type: 'bar',
            stack: '总量',
            label: {
                show: true
            },
            data: [320, 302, 341, 374, 390, 450, 420]
        },
        {
            name: '支出',
            type: 'bar',
            stack: '总量',
            label: {
                show: true,
                position: 'left'
            },
            data: [-120, -132, -101, -134, -190, -230, -210]
        }
    ]
};
```

8.3双向对比柱形图

通过万能组件配置双向对比条形图

案例效果



代码实现

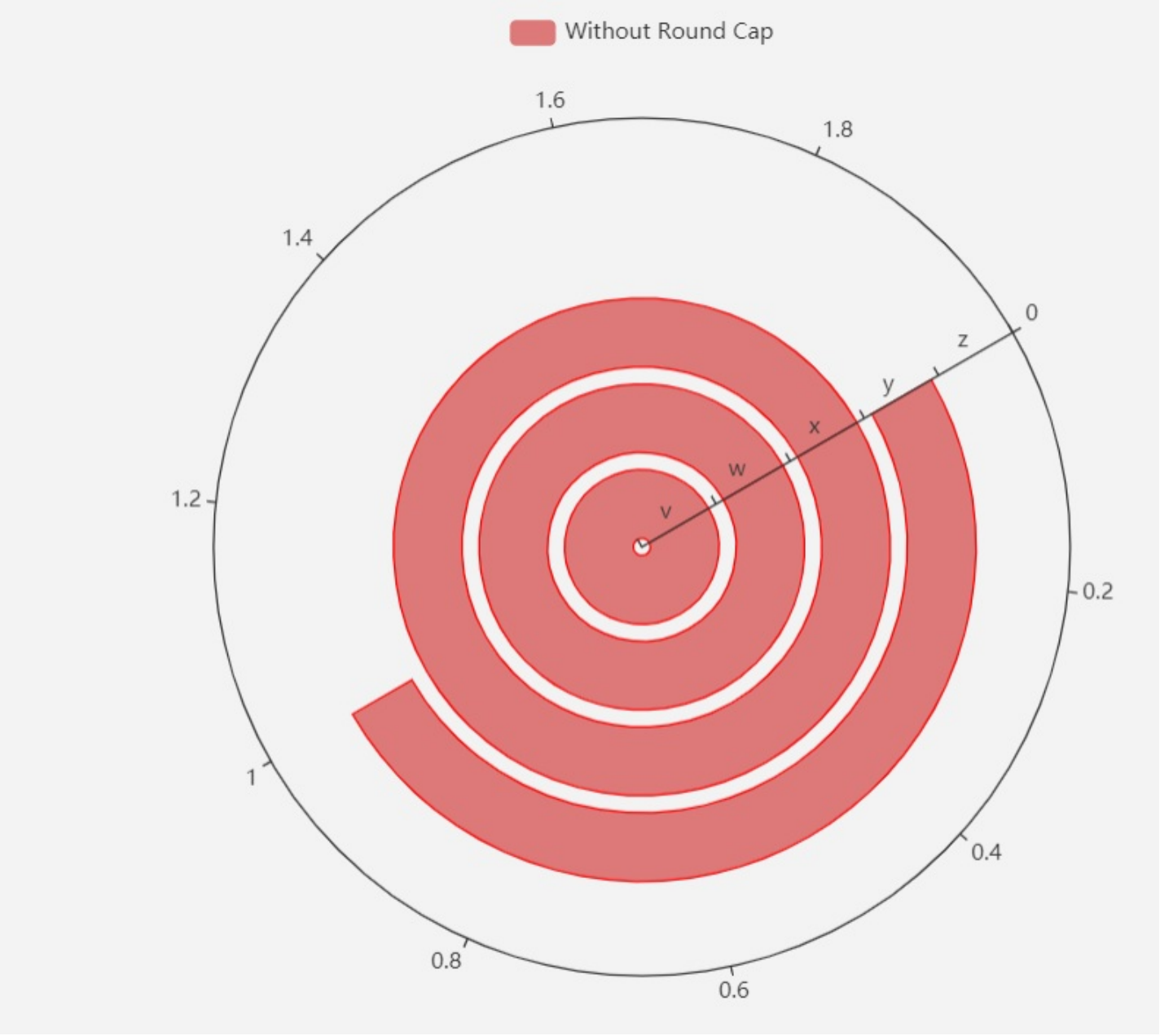
```
option = {
    tooltip: {
        trigger: 'axis',
        axisPointer: {            // 坐标轴指示器，坐标轴触发有效
            type: 'shadow'        // 默认为直线，可选为：'line' | 'shadow'
        }
    },
    legend: {
        data: [ '支出', '收入']
    },
    grid: {
```

```
        left: '3%',
        right: '4%',
        bottom: '3%',
        containLabel: true
    },
    xAxis: [
        {
            type: 'value'
        }
    ],
    yAxis: [
        {
            type: 'category',
            axisTick: {
                show: false
            },
            data: ['周一', '周二', '周三', '周四', '周五', '周六', '周日']
        }
    ],
    series: [

        {
            name: '收入',
            type: 'bar',
            stack: '总量',
            label: {
                show: true
            },
            data: [320, 302, 341, 374, 390, 450, 420]
        },
        {
            name: '支出',
            type: 'bar',
            stack: '总量',
            label: {
                show: true,
                position: 'left'
            },
            data: [-120, -132, -101, -134, -190, -230, -210]
        }
    ]
};
```

8.4圆形柱形图

通过万能组件配置圆形柱形图

案例效果



代码实现

```
option = {
    angleAxis: {
        max: 2,
        startAngle: 30,
        splitLine: {
            show: false
        }
    },
    radiusAxis: {
```

```
        type: 'category',
        data: ['v', 'w', 'x', 'y', 'z'],
        z: 10
    },
    polar: {
    },
    series: [{
        type: 'bar',
        data: [4, 3, 2, 1, 0],
        coordinateSystem: 'polar',
        name: 'Without Round Cap',
        color: 'rgba(200, 0, 0, 0.5)',
        itemStyle: {
            borderColor: 'red',
            borderWidth: 1
        }
    }, ],
    legend: {
        show: true,
        data: ['Without Round Cap']
    }
};
```

8.5嵌套饼图

通过万能组件配置嵌套饼图

案例效果

代码实现

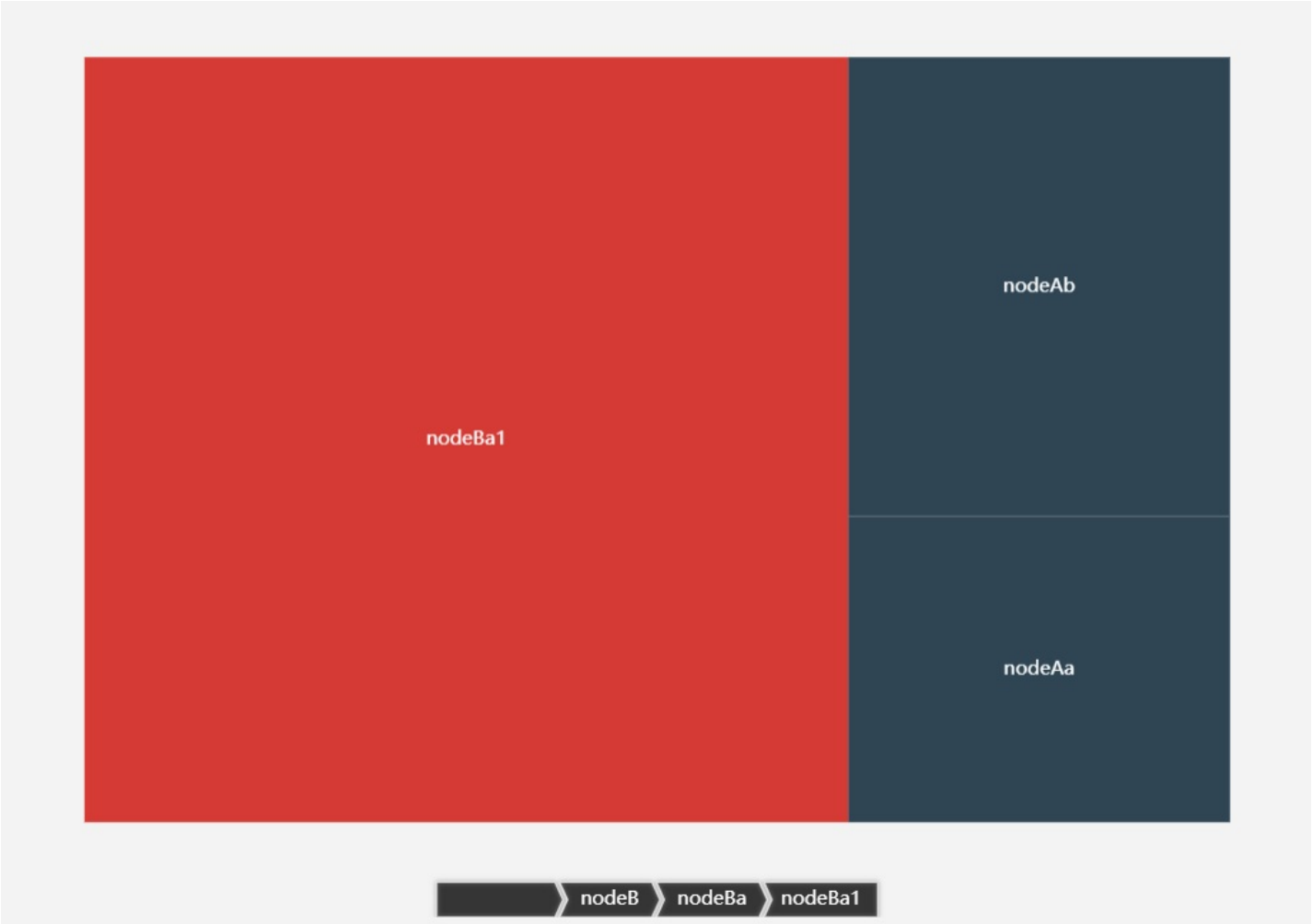
```
option = {
    tooltip: {
        trigger: 'item',
        formatter: '{a} <br/>{b}: {c} ({d}%)'
    },
    legend: {
        orient: 'vertical',
        left: 10,
        data: ['直达', '营销广告', '搜索引擎', '邮件营销', '联盟广告', '视频广告', '百
度', '谷歌', '必应', '其他']
    },
    series: [
        {
            name: '访问来源',
            type: 'pie',
            selectedMode: 'single',
            radius: [0, '30%'],

            label: {
                position: 'inner'
            },
            labelLine: {
                show: false
            },
            data: [
                {value: 335, name: '直达', selected: true},
                {value: 679, name: '营销广告'},
                {value: 1548, name: '搜索引擎'}
            ]
        },
        {
            name: '访问来源',
            type: 'pie',
            radius: ['40%', '55%'],
            label: {
                formatter: '{a|{a}}{abg|}\n{hr|}\n  {b|{b}：}{c}  {per|{d}%}  '
,
                backgroundColor: '#eee',
                borderColor: '#aaa',
```

```
                borderWidth: 1,
                borderRadius: 4,
                rich: {
                    a: {
                        color: '#999',
                        lineHeight: 22,
                        align: 'center'
                    },

                    hr: {
                        borderColor: '#aaa',
                        width: '100%',
                        borderWidth: 0.5,
                        height: 0
                    },
                    b: {
                        fontSize: 16,
                        lineHeight: 33
                    },
                    per: {
                        color: '#eee',
                        backgroundColor: '#334455',
                        padding: [2, 4],
                        borderRadius: 2
                    }
                }
            },
            data: [
                {value: 335, name: '直达'},
                {value: 310, name: '邮件营销'},
                {value: 234, name: '联盟广告'},
                {value: 135, name: '视频广告'},
                {value: 1048, name: '百度'},
                {value: 251, name: '谷歌'},
                {value: 147, name: '必应'},
                {value: 102, name: '其他'}
            ]
        }
    ]
};
```

8.6矩形树图

通过万能组件配置矩形树图

案例效果



代码实现

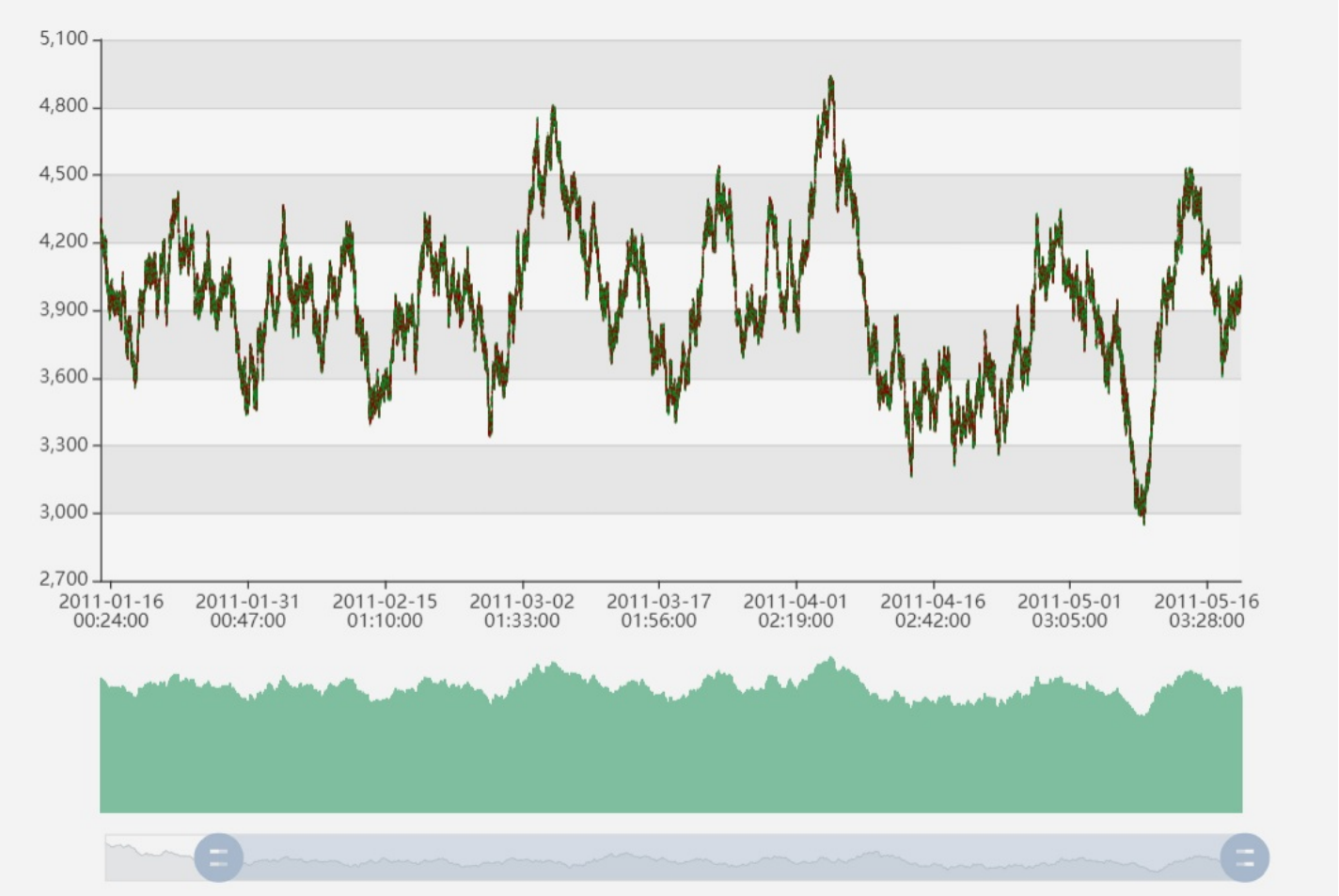
```
option = {
    series: [{
        type: 'treemap',
        data: [{
            name: 'nodeA',              // First tree
            value: 10,
            children: [{
                name: 'nodeAa',         // First leaf of first tree
                value: 4
            }, {
                name: 'nodeAb',         // Second leaf of first tree
                value: 6
            }]
        }, {
            name: 'nodeB',              // Second tree
```

```
            value: 20,
            children: [{
                name: 'nodeBa',       // Son of first tree
                value: 20,
                children: [{
                    name: 'nodeBa1',  // Granson of first tree
                    value: 20
                }]
            }]
        }]
    }]
};
```

8.7k线图

通过万能组件配置k线图

案例效果



代码实现

```
var upColor = '#ec0000';
var upBorderColor = '#8A0000';
var downColor = '#00da3c';
var downBorderColor = '#008F28';

var dataCount = 2e5;
var data = generateOHLC(dataCount);

var option = {
    dataset: {
        source: data
    },
    title: {
        text: 'Data Amount: ' + echarts.format.addCommas(dataCount)
    },
    tooltip: {
        trigger: 'axis',
```

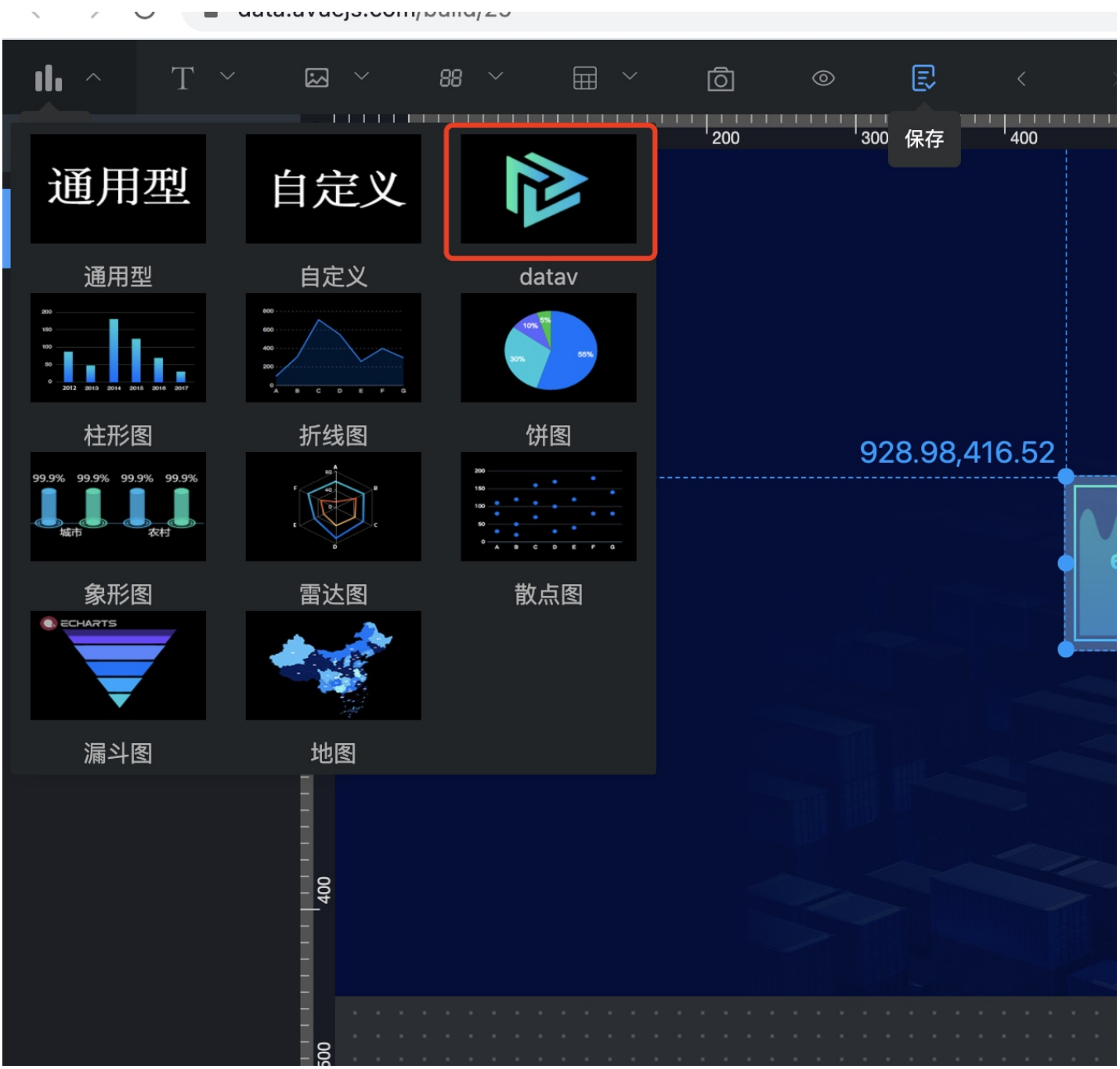
```
        axisPointer: {
            type: 'line'
        }
    },
    toolbox: {
        feature: {
            dataZoom: {
                yAxisIndex: false
            },
        }
    },
    grid: [
        {
            left: '10%',
            right: '10%',
            bottom: 200
        },
        {
            left: '10%',
            right: '10%',
            height: 80,
            bottom: 80
        }
    ],
    xAxis: [
        {
            type: 'category',
            scale: true,
            boundaryGap: false,
            // inverse: true,
            axisLine: {onZero: false},
            splitLine: {show: false},
            splitNumber: 20,
            min: 'dataMin',
            max: 'dataMax'
        },
        {
            type: 'category',
            gridIndex: 1,
            scale: true,
            boundaryGap: false,
            axisLine: {onZero: false},
            axisTick: {show: false},
            splitLine: {show: false},
            axisLabel: {show: false},
            splitNumber: 20,
            min: 'dataMin',
            max: 'dataMax'
        }
    ],
```

```
    yAxis: [
        {
            scale: true,
            splitArea: {
                show: true
            }
        },
        {
            scale: true,
            gridIndex: 1,
            splitNumber: 2,
            axisLabel: {show: false},
            axisLine: {show: false},
            axisTick: {show: false},
            splitLine: {show: false}
        }
    ],
    dataZoom: [
        {
            type: 'inside',
            xAxisIndex: [0, 1],
            start: 10,
            end: 100
        },
        {
            show: true,
            xAxisIndex: [0, 1],
            type: 'slider',
            bottom: 10,
            start: 10,
            end: 100,
            handleIcon: 'M10.7,11.9H9.3c-4.9,0.3-8.8,4.4-8.8,9.4c0,5,3.9,9.1,8.
8,9.4h1.3c4.9-0.3,8.8-4.4,8.8-9.4C19.5,16.3,15.6,12.2,10.7,11.9z M13.3,24.4H6.7
V23h6.6V24.4z M13.3,19.6H6.7v-1.4h6.6V19.6z',
            handleSize: '105%'
        }
    ],
    visualMap: {
        show: false,
        seriesIndex: 1,
        dimension: 6,
        pieces: [{
            value: 1,
            color: upColor
        }, {
            value: -1,
            color: downColor
        }]
    },
    series: [
```

```
        {
            type: 'candlestick',
            itemStyle: {
                color: upColor,
                color0: downColor,
                borderColor: upBorderColor,
                borderColor0: downBorderColor
            },
            encode: {
                x: 0,
                y: [1, 4, 3, 2]
            }
        },
        {
            name: 'Volumn',
            type: 'bar',
            xAxisIndex: 1,
            yAxisIndex: 1,
            itemStyle: {
                color: '#7fbe9e'
            },
            large: true,
            encode: {
                x: 0,
                y: 5
            }
        }
    ]
};

function generateOHLC(count) {
    var data = [];

    var xValue = +new Date(2011, 0, 1);
    var minute = 60 * 1000;
    var baseValue = Math.random() * 12000;
    var boxVals = new Array(4);
    var dayRange = 12;

    for (var i = 0; i < count; i++) {
        baseValue = baseValue + Math.random() * 20 - 10;

        for (var j = 0; j < 4; j++) {
            boxVals[j] = (Math.random() - 0.5) * dayRange + baseValue;
        }
        boxVals.sort();

        var openIdx = Math.round(Math.random() * 3);
        var closeIdx = Math.round(Math.random() * 2);
        if (closeIdx === openIdx) {
```

```
            closeIdx++;
        }
        var volumn = boxVals[3] * (1000 + Math.random() * 500);

        // ['open', 'close', 'lowest', 'highest', 'volumn']
        // [1, 4, 3, 2]
        data[i] = [
            echarts.format.formatTime('yyyy-MM-dd\nhh:mm:ss', xValue += minute)
,
            +boxVals[openIdx].toFixed(2), // open
            +boxVals[3].toFixed(2), // highest
            +boxVals[0].toFixed(2), // lowest
            +boxVals[closeIdx].toFixed(2),  // close
            volumn.toFixed(0),
            getSign(data, i, +boxVals[openIdx], +boxVals[closeIdx], 4) // sign
        ];
    }

    return data;

    function getSign(data, dataIndex, openVal, closeVal, closeDimIdx) {
        var sign;
        if (openVal > closeVal) {
            sign = -1;
        }
        else if (openVal < closeVal) {
            sign = 1;
        }
        else {
            sign = dataIndex > 0
                // If close === open, compare with close of last record
                ? (data[dataIndex - 1][closeDimIdx] <= closeVal ? 1 : -1)
                // No record of previous, set to be positive
                : 1;
        }

        return sign;
    }
}
```

9万能dataV组件



一、组件名称设置

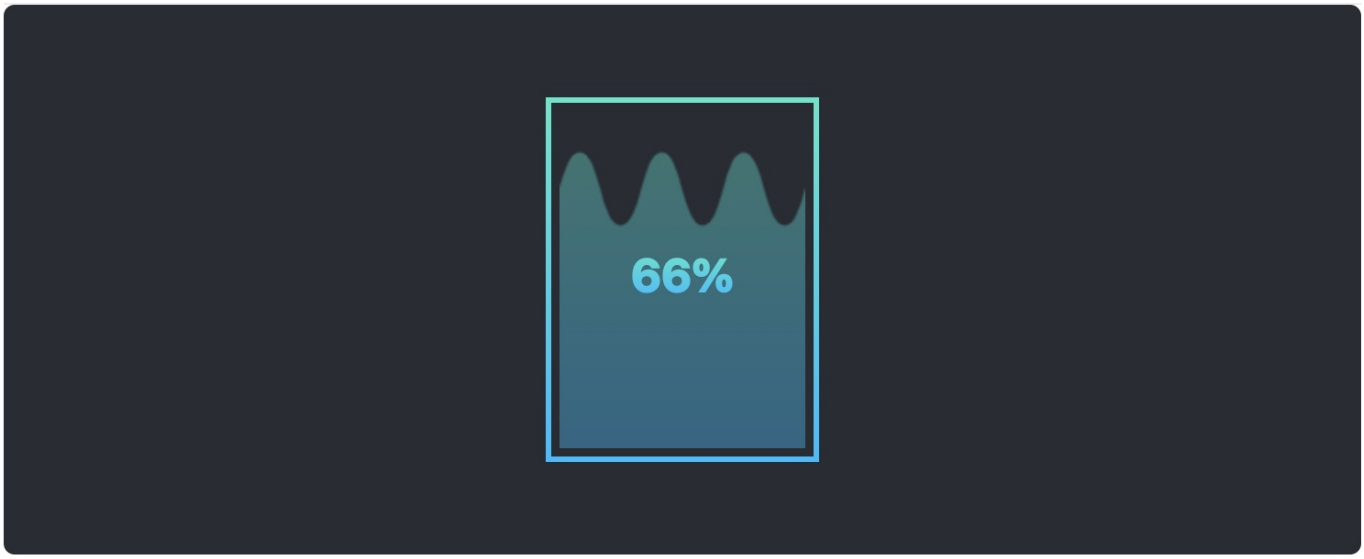

操作
datav
图层名称
datav
隐藏
在线文档
点击查看
模块名称
dv-water-level-pond
配置列表
编辑

设置组件的数据格式，可以参考官网的config配置http://datav.jiaminghi.com/guide/waterLevelPond.html

数据处理

```
(data)=>{
  console.log(data);
  return {
    config:{
      data: [66]
    }
  }
}
```

矩形水位图



点击以展示/隐藏config数据

```js
export default {
  data: [66]
}
```

二、接口设置

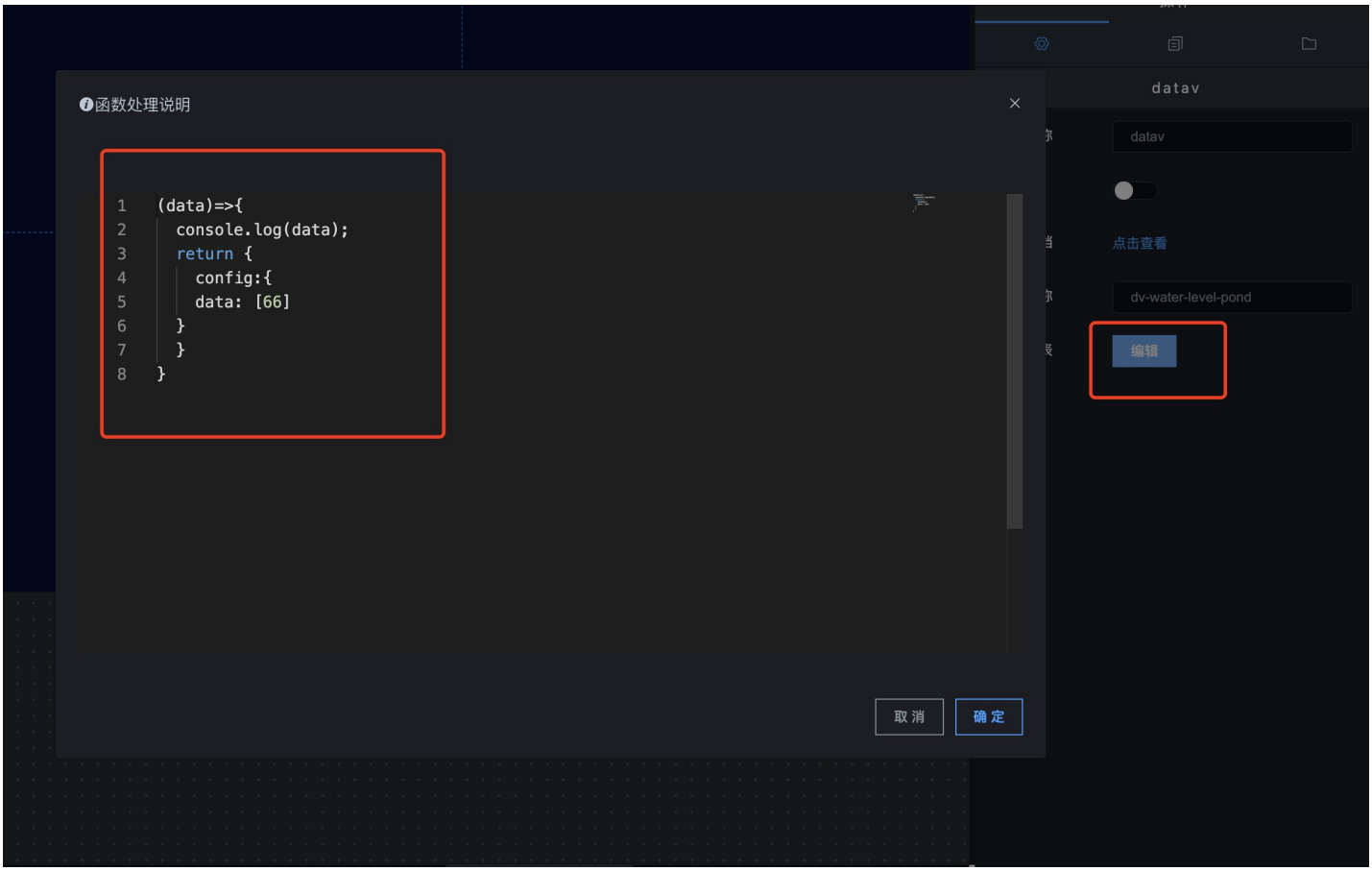
1. 数据类型

数据类型分为静态数据和动态数据；

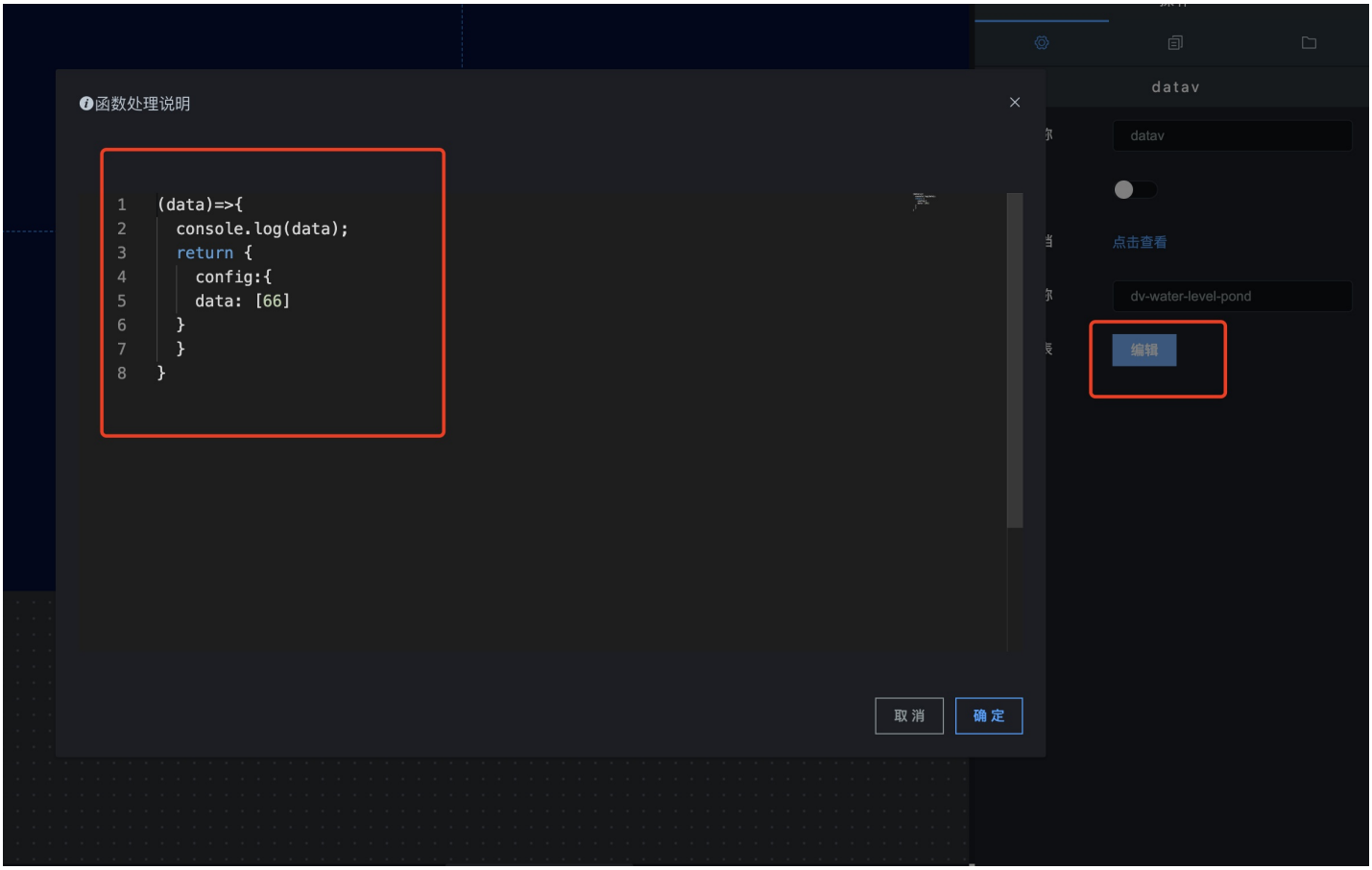
- 静态数据：写死的数据；
- 动态数据：会随着接口传过来的数据实时变化；

2. 接口地址

（1）静态数据，和组件结构一致



（2）动态数据，和组件结构一致



3. 刷新时间

这个参数主要针对动态数据设置的，完成数据的实时更新。

- 如果你想设置成5秒刷新一次，可以将刷新时间设置成“5000”；

自定义Vue组件

> 可以在线自定义Vue组件和引入三方库，文中例子以结合layer弹窗为例
>
> 点击我跳转参考例子


data.avuejs.com/build/28
通用型
通用型
datav
柱形图
折线图
饼图
象形图
雷达图
散点图
漏斗图
矩形图
定时器
自定义
自定义Vue组件

```
<template>
    <div class = "test">
        <el - button type = "success"@ click = "handleClick" > 点击我弹窗 < /el-
button>
        <h1>{{dataChart}}<h1>
    </div>
</template>
<script>
export default{
    data(){
        return{}
    },
    created(){ },
    methods:{
        handleClick(){
            this.$message.success(this.dataChart.name)
            window.$loadScript('js','https:/ / cdn.staticfile.org / jquery / 2.
1.2 / jquery.js').then(()=>{
                return window.$loadScript('js ','https: //data.avuejs.com/layer
/layer.js')
            }).then(() = > {
                return window.$loadScript('js','https://cdn.staticfile.org/jque
ry.qrcode/1.0/jquery.qrcode.min.js')
            }).then(() = > {
                    layer.open({
                        title: '生成二维码',
                        content: '<div id="qrcode"></div>',
                        success: function(layero, index) {
                            var config = {
                                width: 200,
                                height: 200,
                                text: "苦逼的程序员"
                            }
                            $("#qrcode").qrcode(config);
                        }
                    });
            })
        }
    }
}
< /script>
<style>
    .test{
        text-align:center;
        color:red;
        font-size:40px;
    }
</style >
```

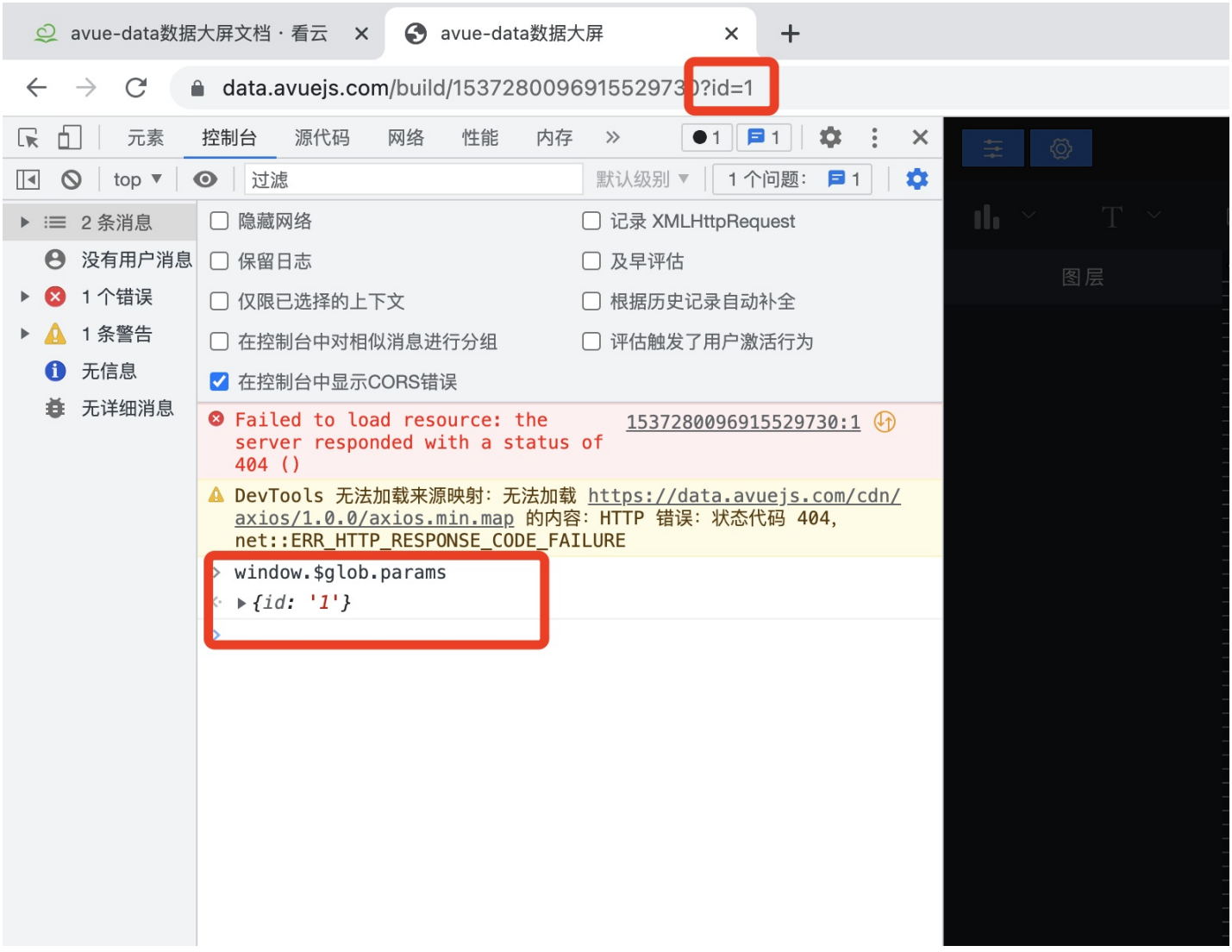
全局变量

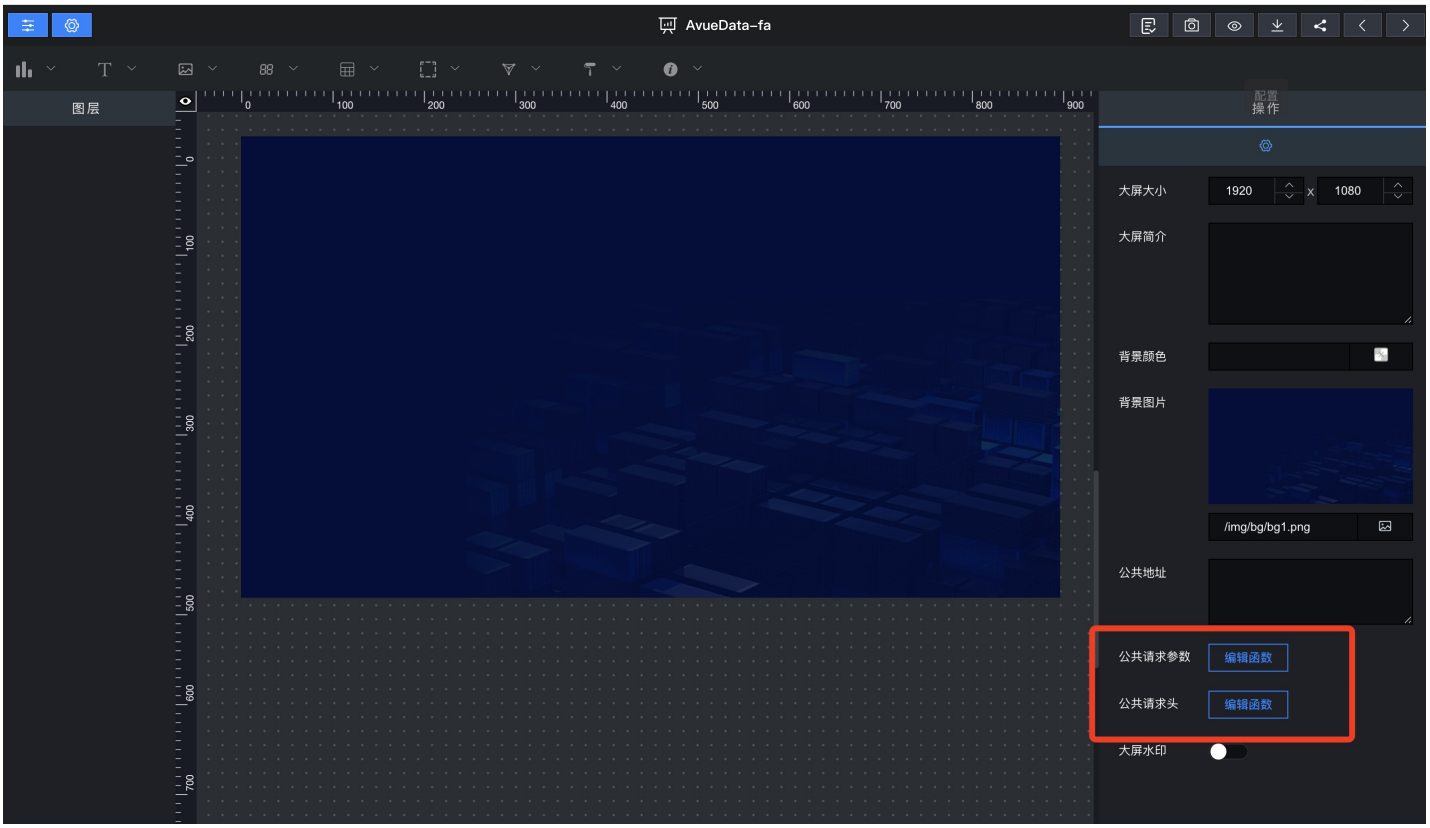
> 全局变量，它可以作用于axios发送数据中

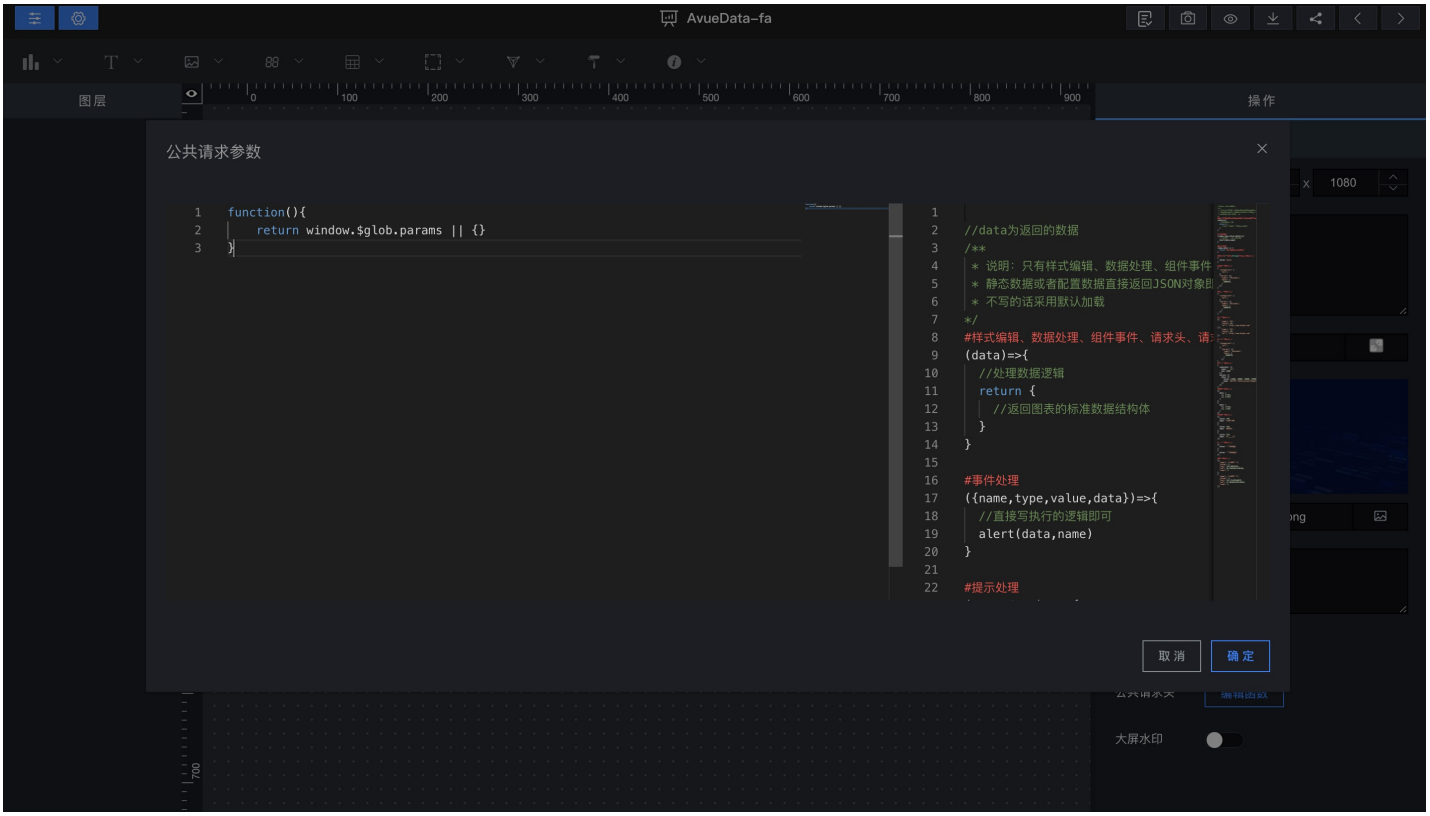
作用域

- header头部
- body发送体
- url参数后面

他可以手动配置，也可以自动获取window.$glob全局变量下的参数




AvueData-fa
图层
操作
大屏大小
1920
1080
大屏简介
背景颜色
背景图片
/img/bg/bg1.png
公共地址
公共请求参数
编辑函数
公共请求头
编辑函数
大屏水印

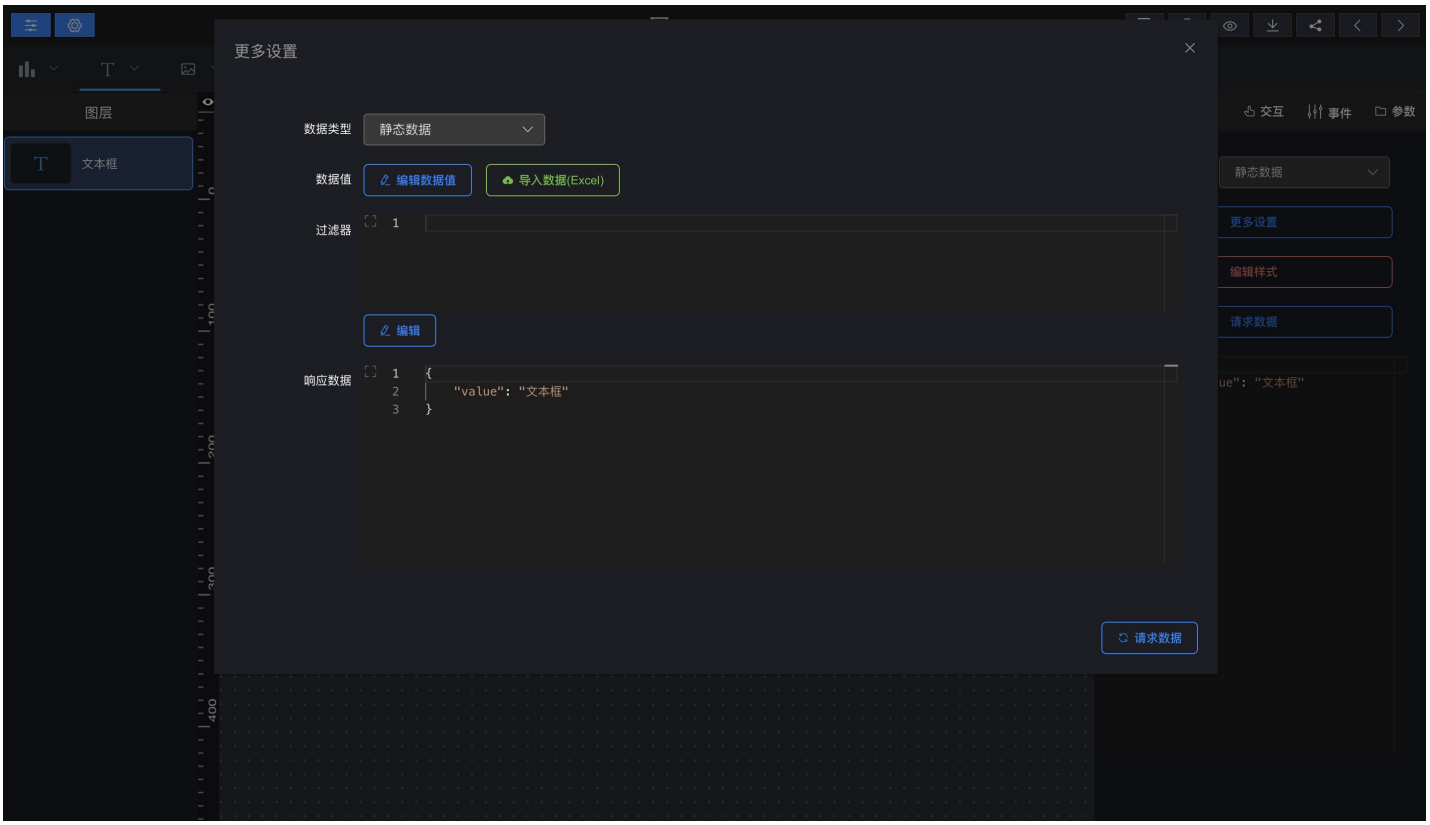

AvueData-fa
图层
操作
公共请求参数
function(){
return window.$glob.params || {}
}
//data为返回的数据
/**
* 说明：只有样式编辑、数据处理、组件事件
* 静态数据或者配置数据直接返回JSON对象即
* 不写的话采用默认加载
*/
#样式编辑、数据处理、组件事件、请求头、请
(data)=>{
//处理数据逻辑
return {
//返回图表的标准数据结构体
}
}
#事件处理
({name,type,value,data})=>{
//直接写执行的逻辑即可
alert(data,name)
}
#提示处理
取 消
确 定
大屏水印

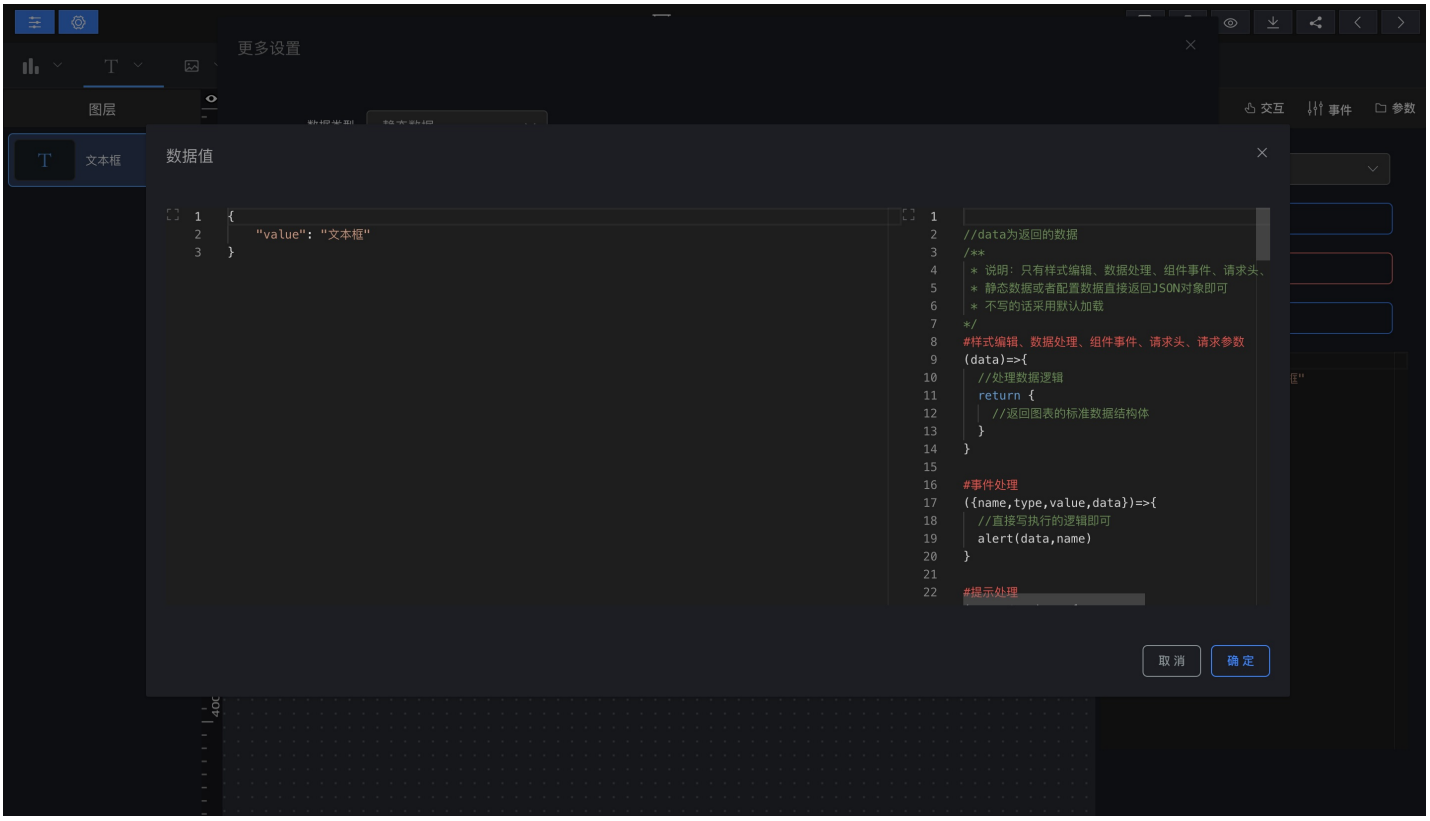
组件数据交互

> 组件请求回数据、完后返回组件可以识别的标准格式即可。

一、静态数据

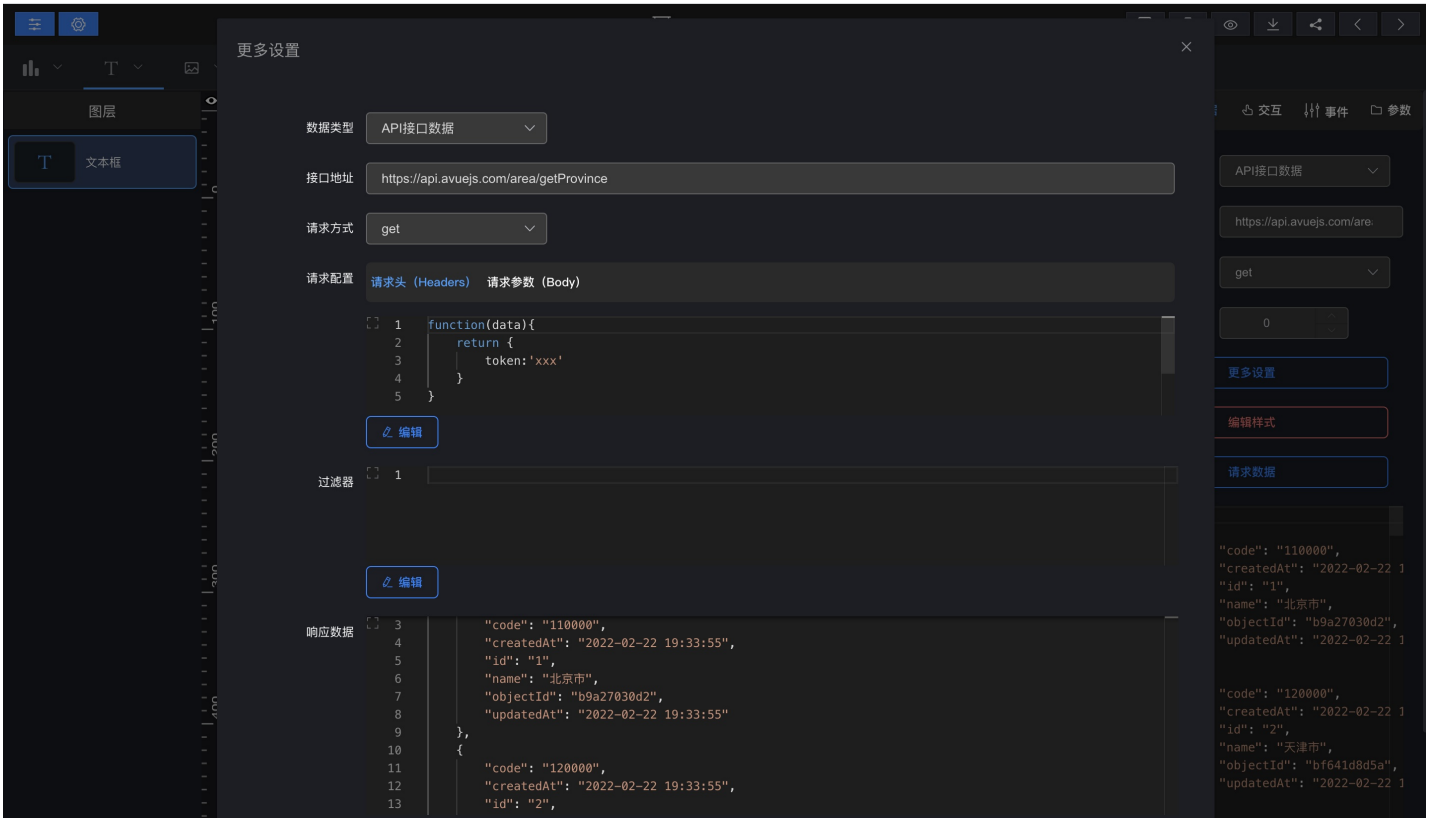
直接配置成组件对应的数据格式即可





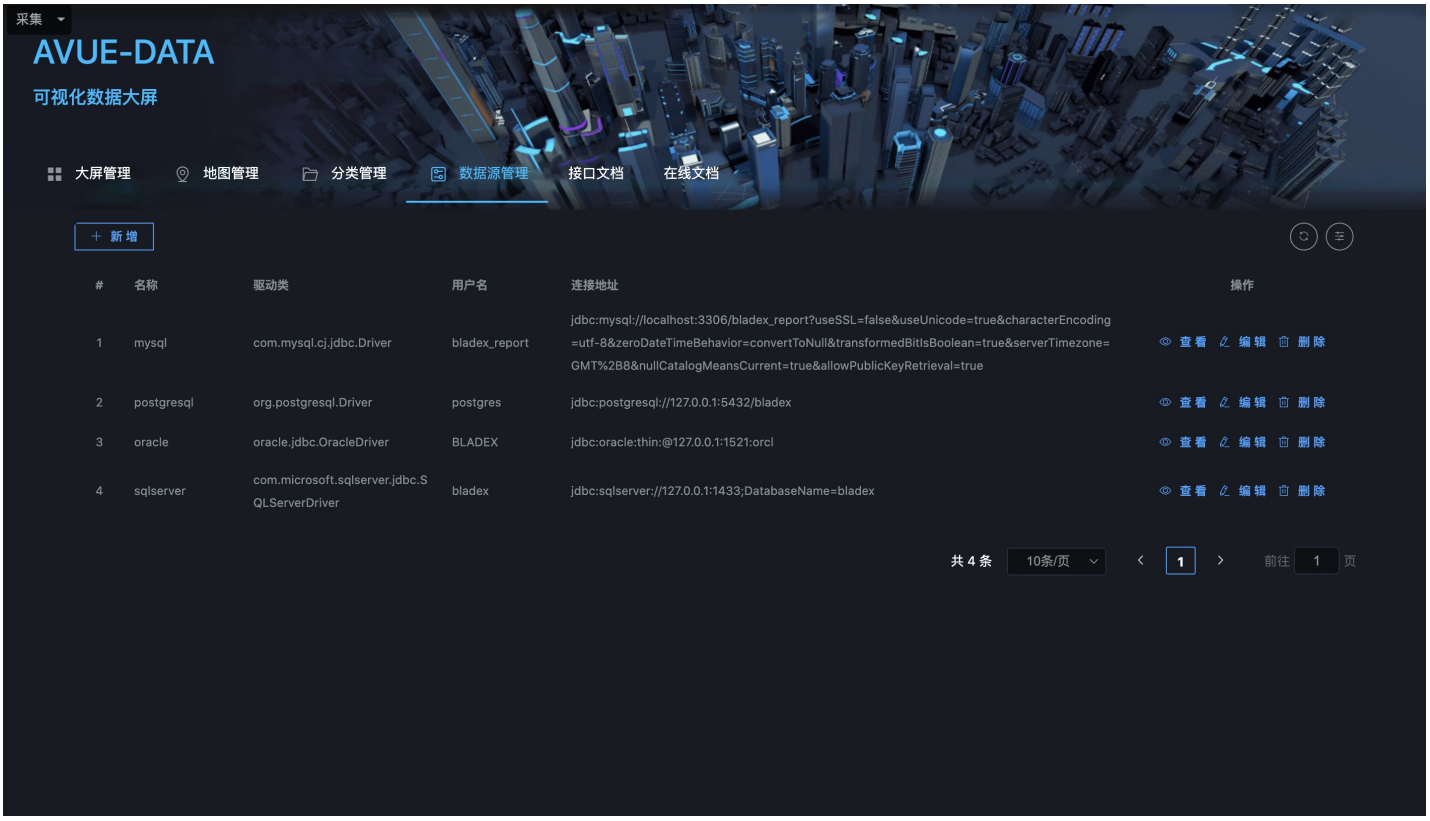
二、API接口数据

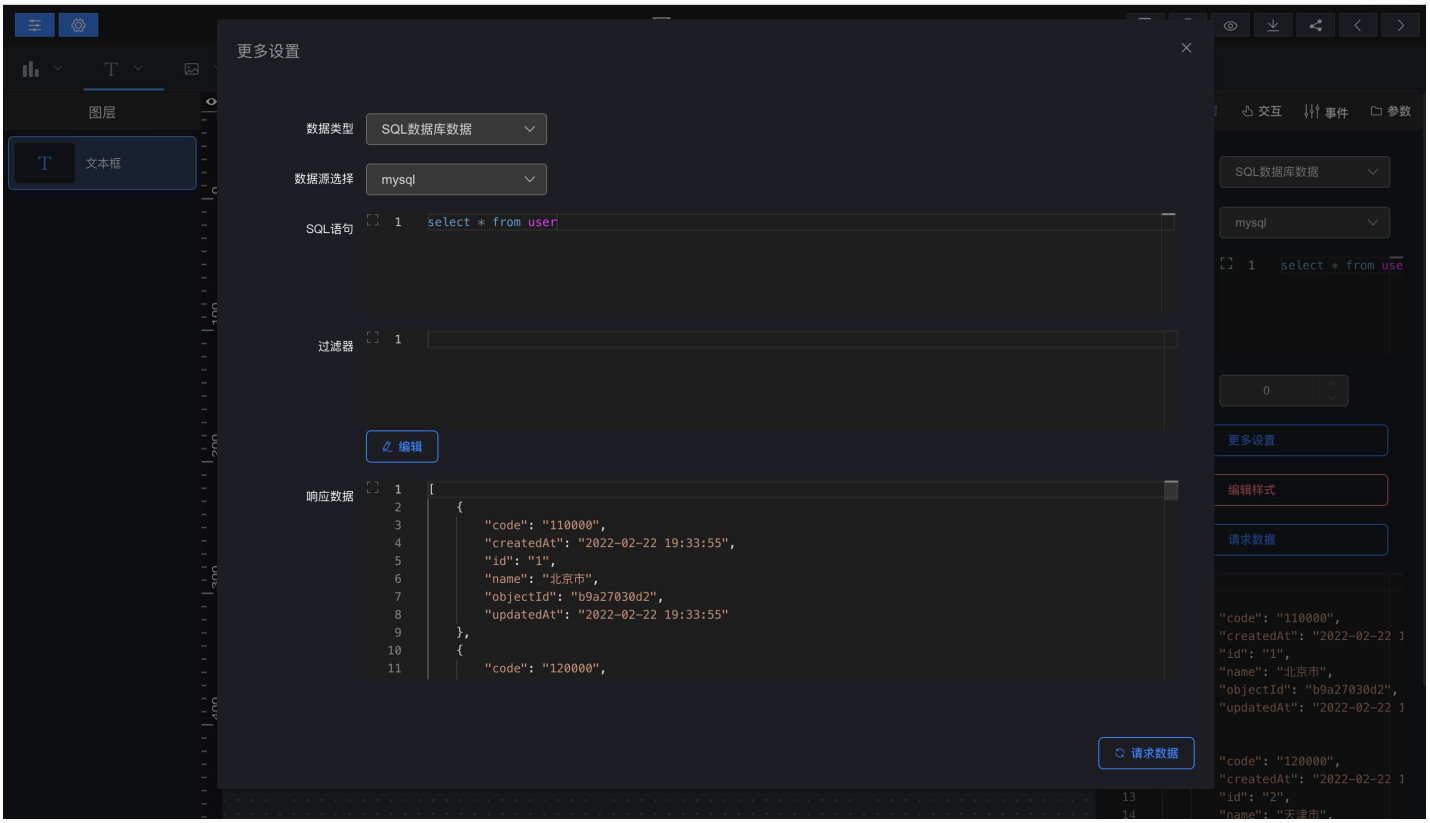
填写API地址，配置相关请求参数，返回的数据处理成组件对应的数据格式即可



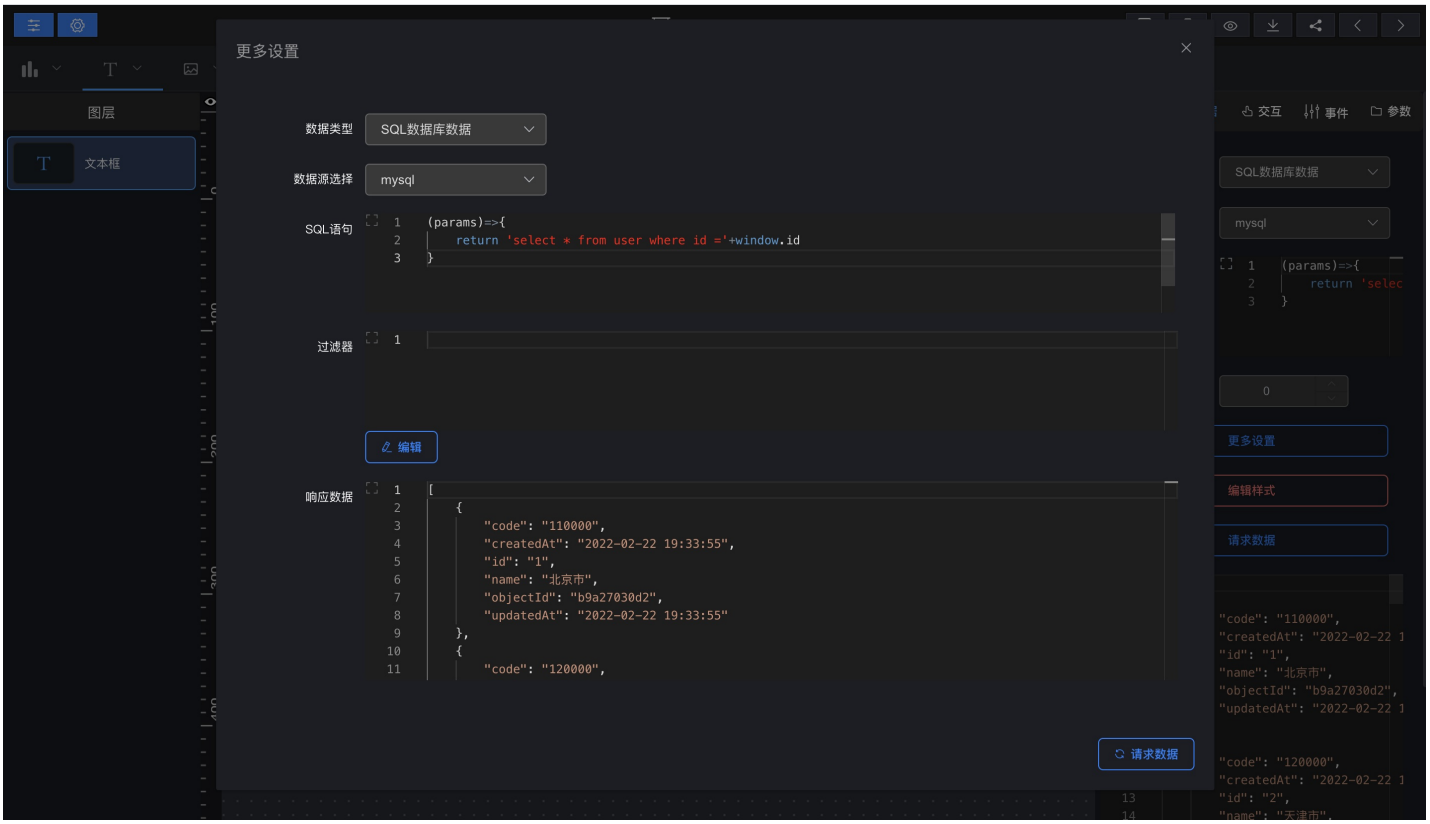
三、SQL数据源数据

配数据源，编写对应的sql语法即可，数据格式处理和API一样



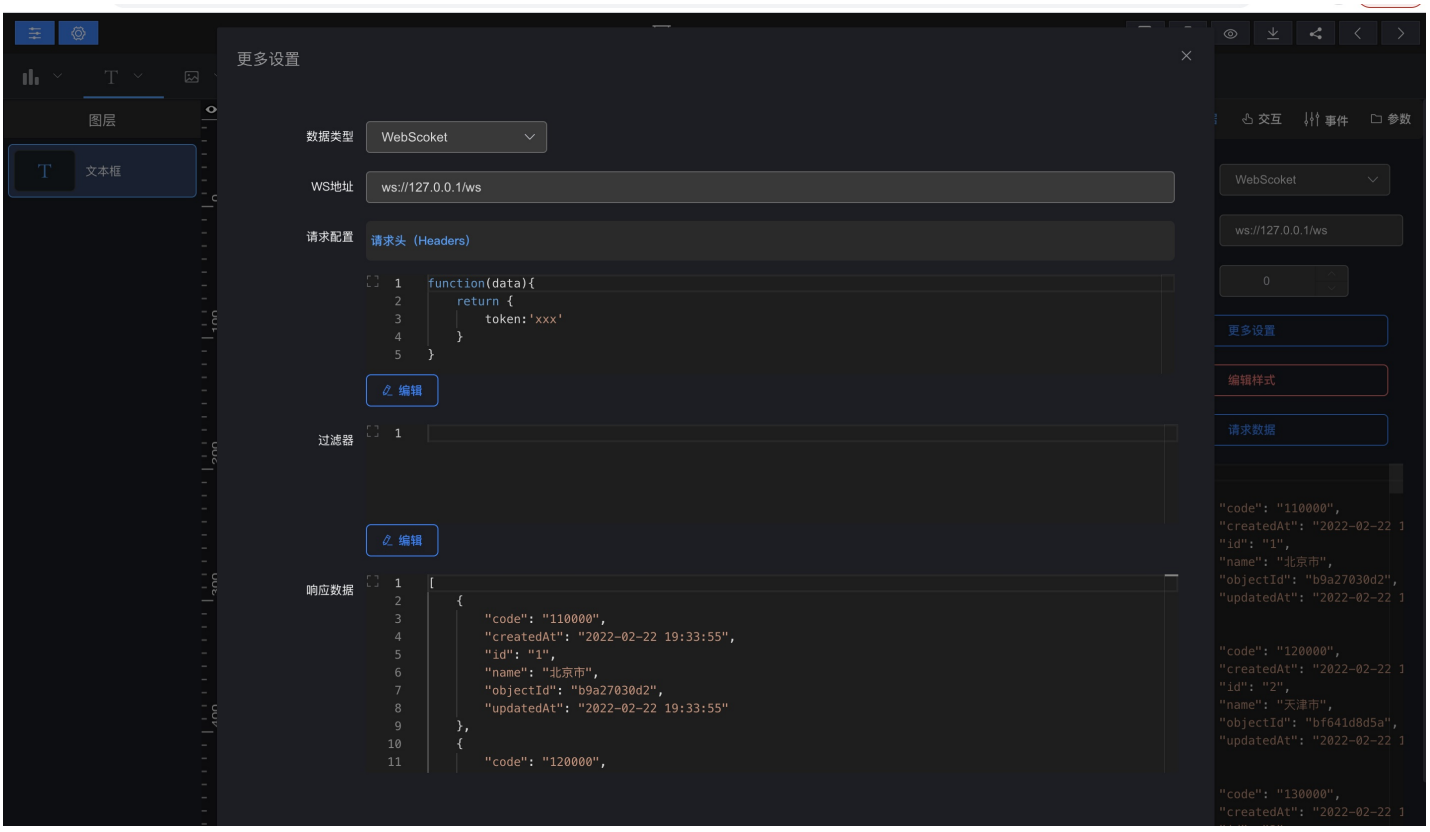


这里的SQl是可以配置成js函数的，这样就可以取各种参数,处理后返回最终的sql语句即可



四、Websoket长链接数据

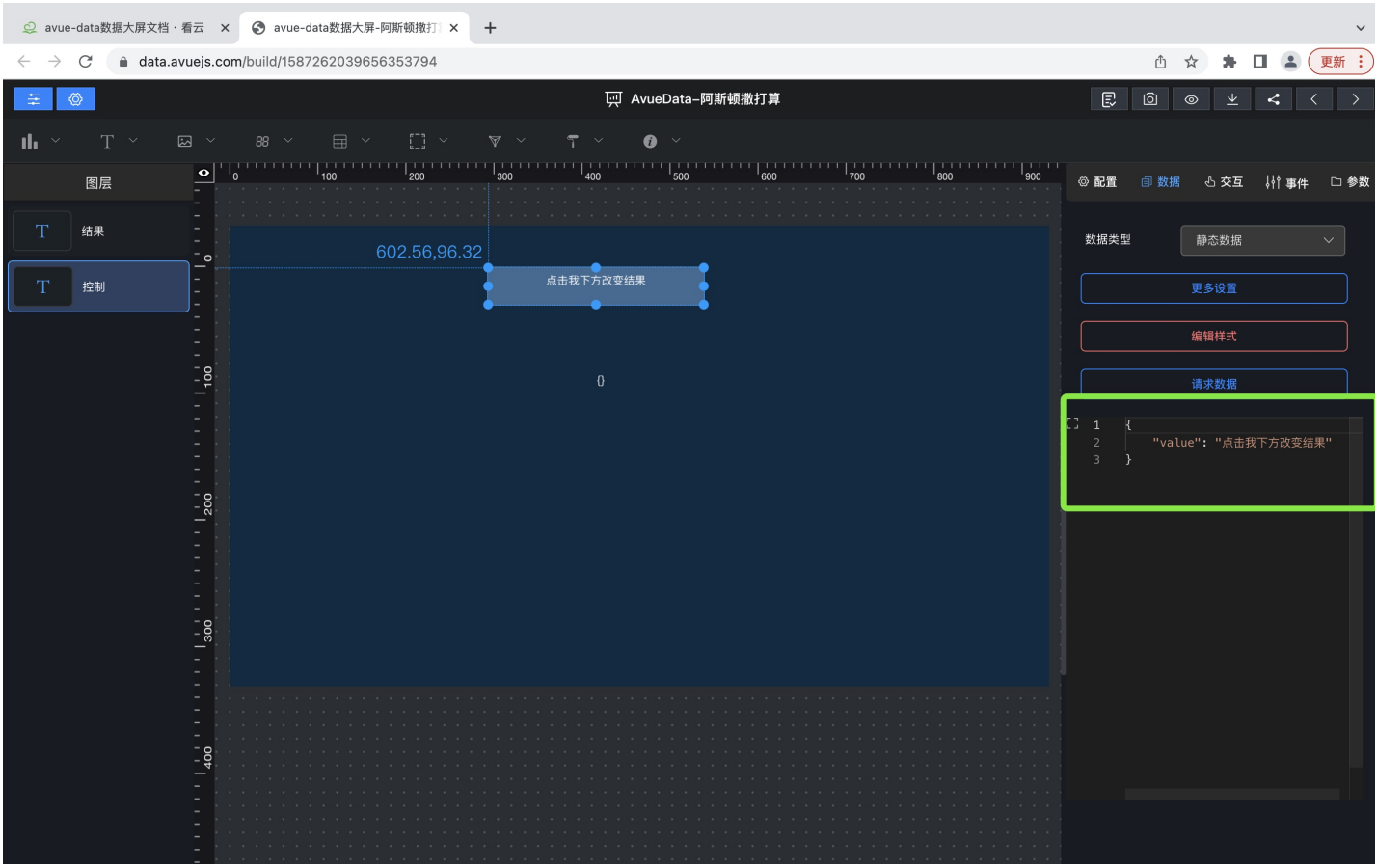
配置场链接地址即可返回数据

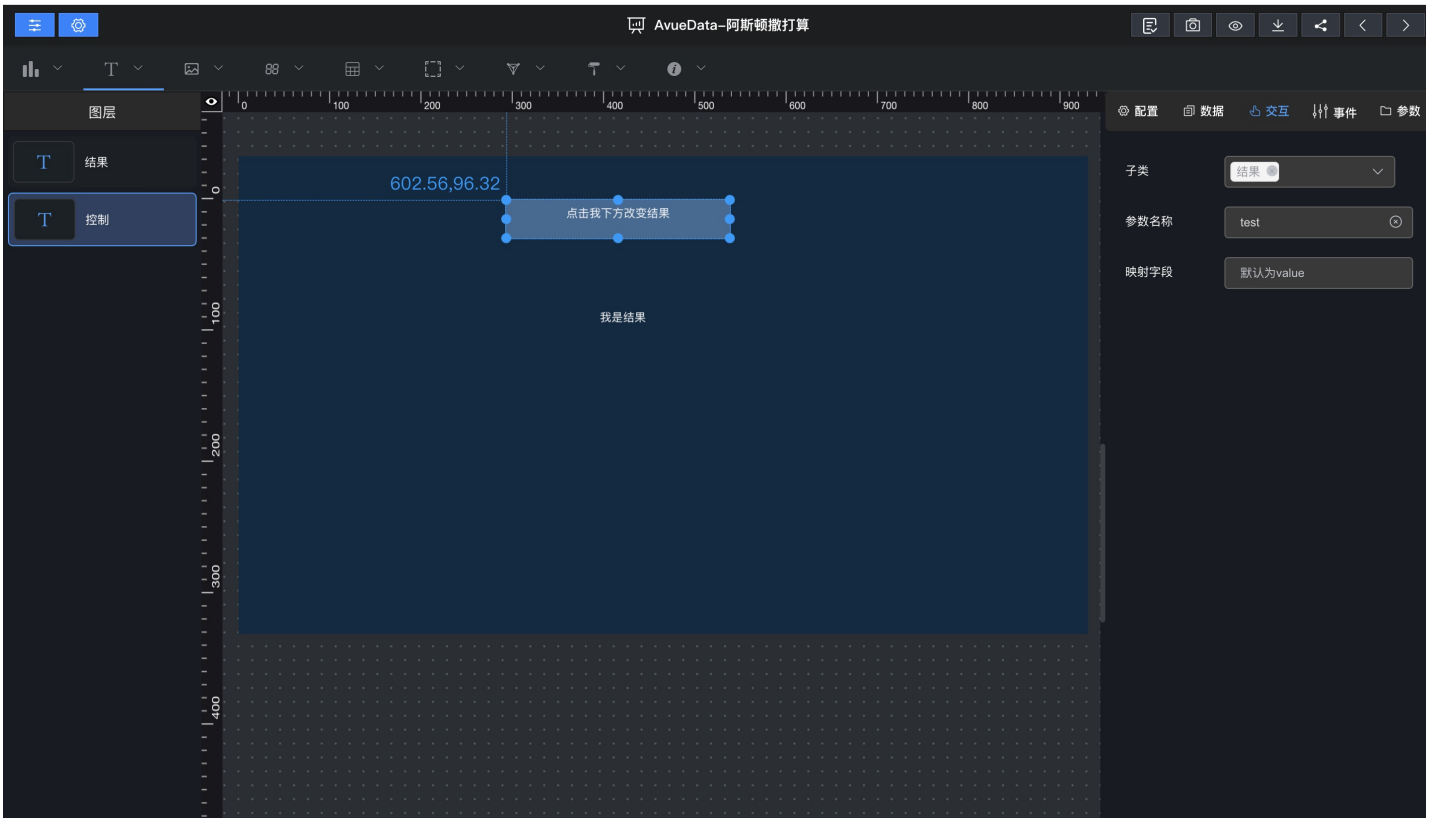


组件参数交互

> 组件之间的参数交互

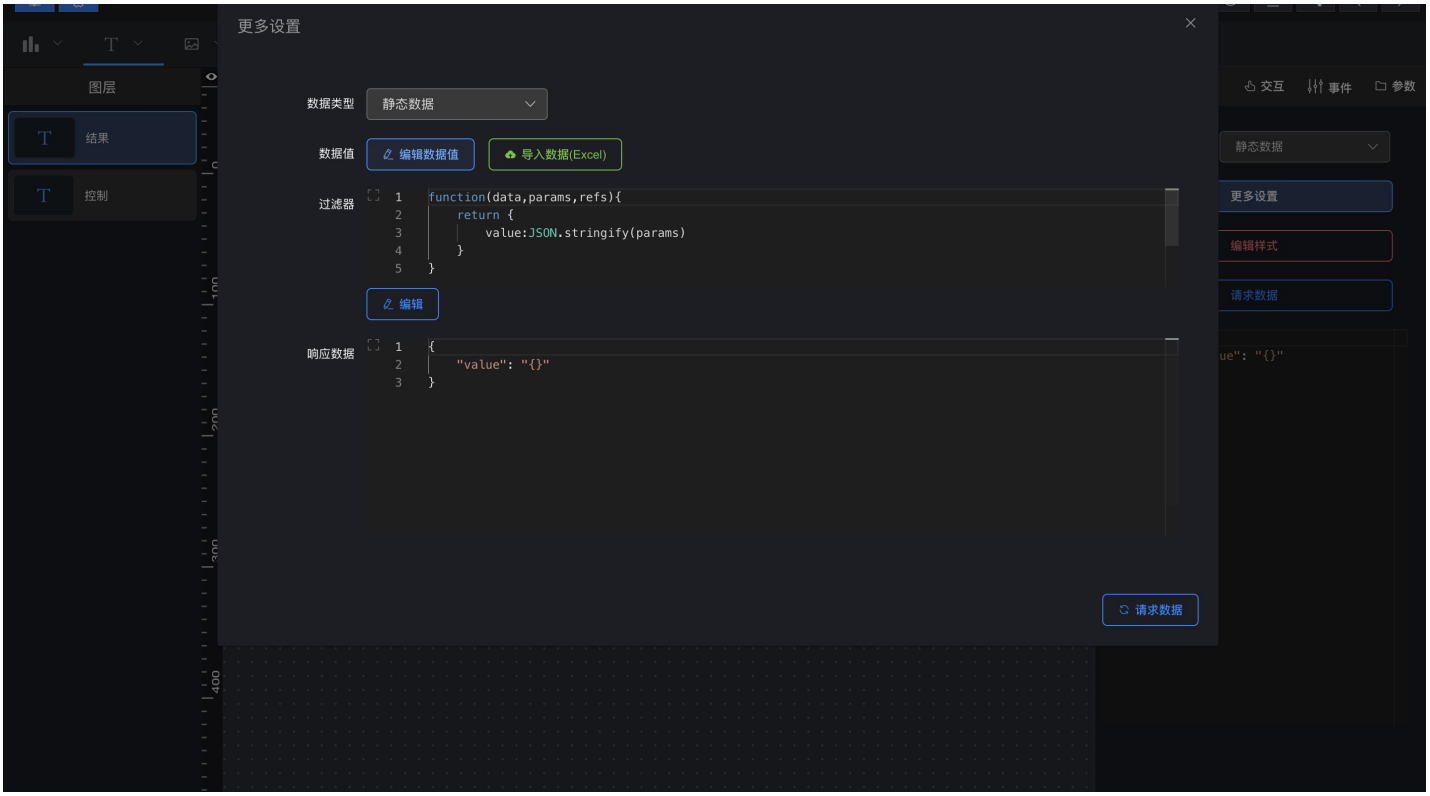
选中你要交互的组件(可以是多个)，并且填写传递过去参数的名称，映射字段就是取你数据中的那个字段，默认为value





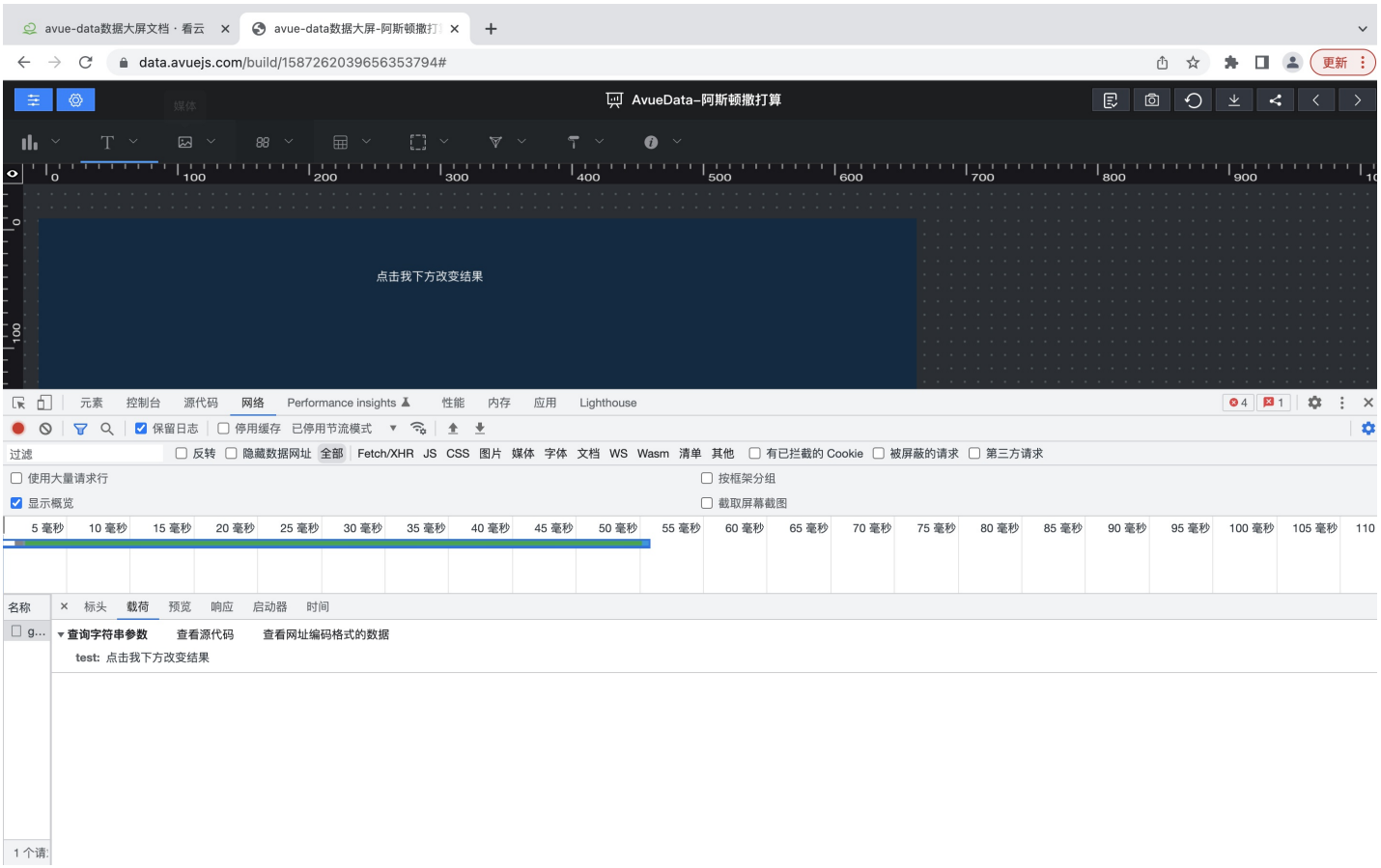
一、静态数据

可以在过滤器中去处理，过滤器中的params参数就是传递过来的参数，根据参数处理结果后返回即可



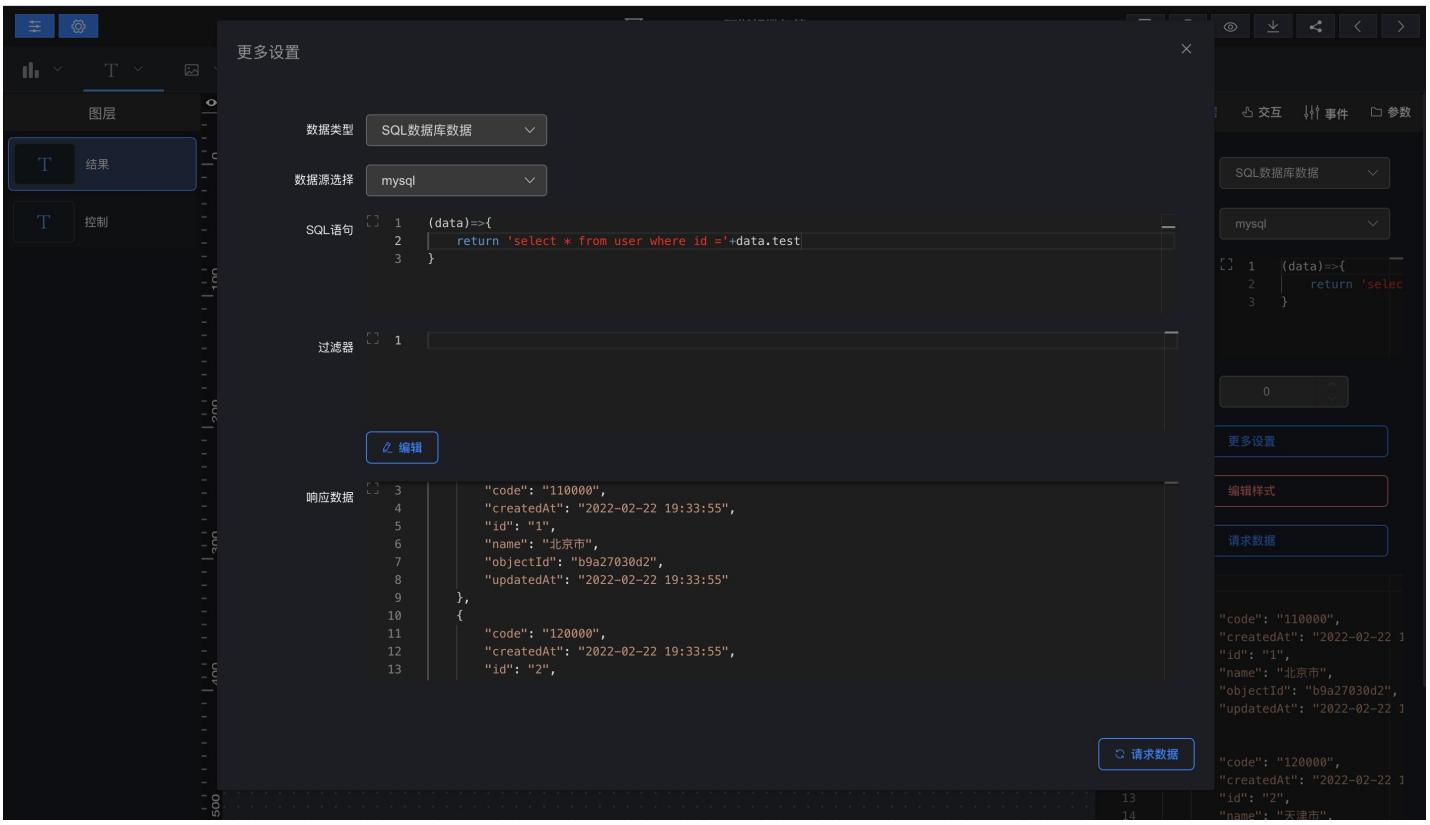
二、API接口数据

不需要在过滤器中去处理结果的，他会根据你的请求，自定把参数帮你带入



三、SQL数据源数据

sql语句可以是字符串也可以是一个函数，函数中的data字段就是传递过来的数据，用来组装你的sql语句



四、Websoket长链接数据

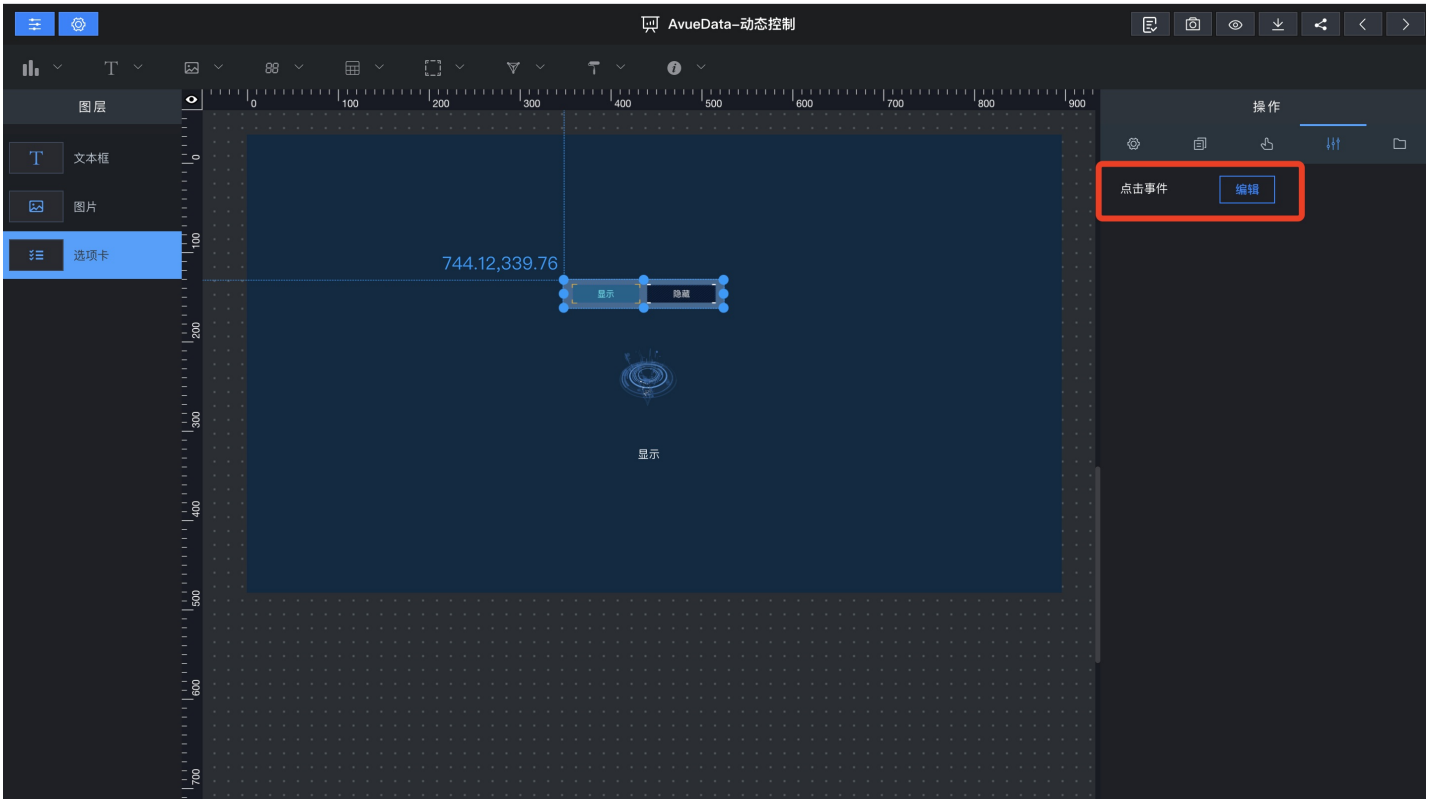
这里和API接口一样，会帮你把参数带入请求中

组件联动交互

> 组件之间的动态联动

[点击我跳转参考例子]

新建组件找到对应的交互事件地方

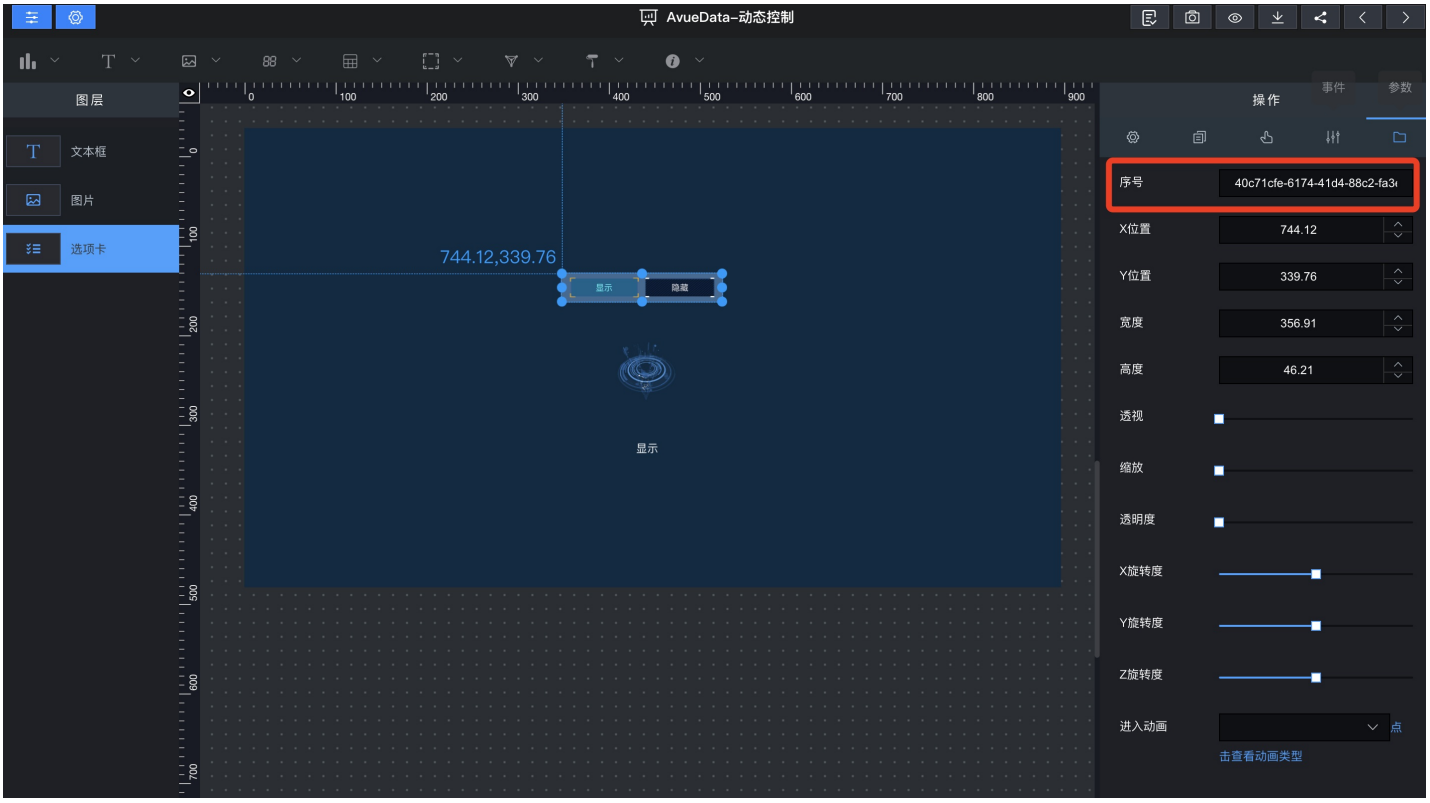


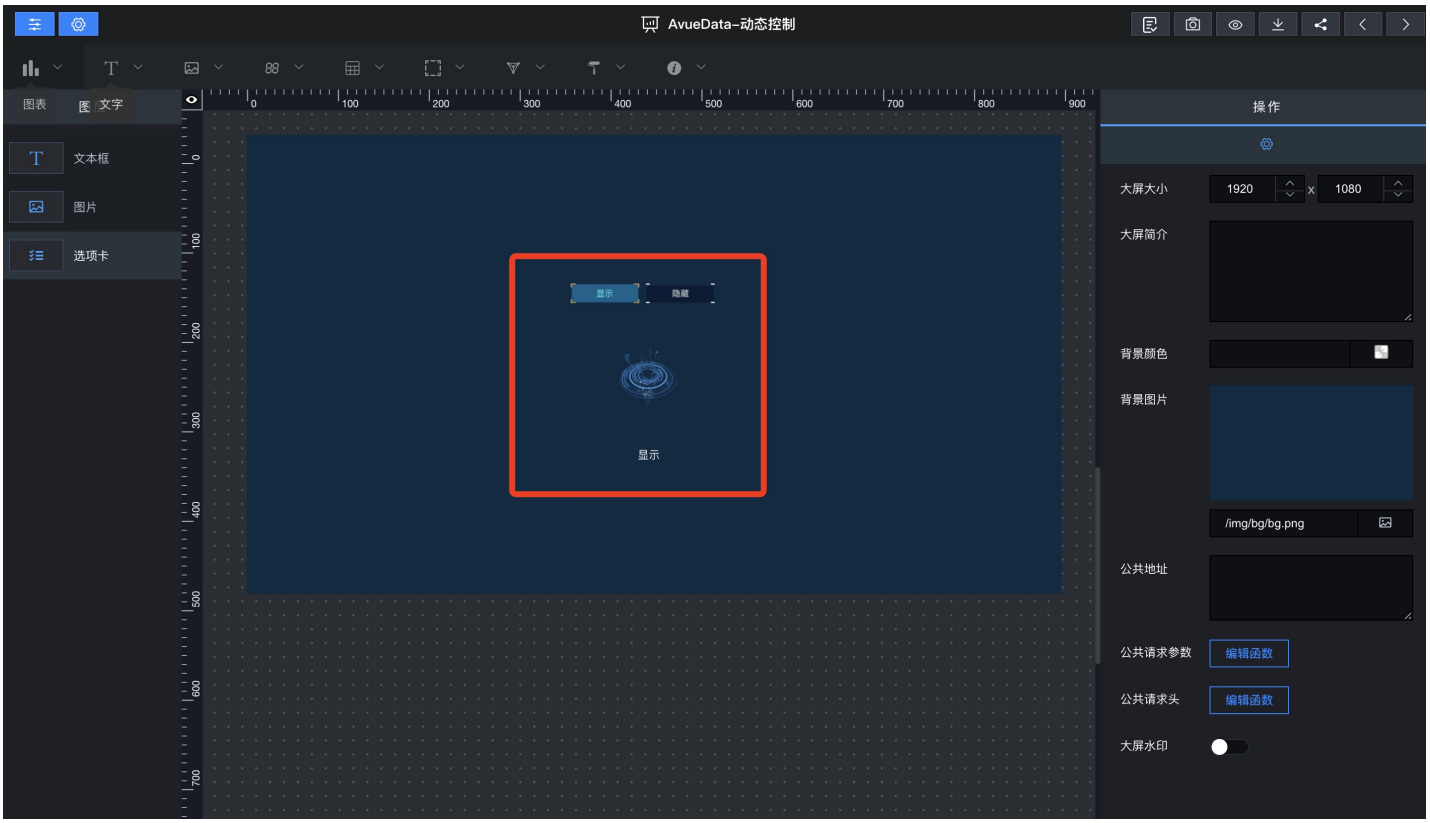
这里可JS处理逻辑，同时也可以拿到window.$glob全局变量

- params参数为点击时候的数据
- refs为画布中全部组件对象，其中ID就是组件的对应序号

```
function(params,refs){

}
```

比如我们做一个动态显隐和赋值，画布中新建3个对应组件，并且找到他们的ID






```
function(params,refs){
    #图片组件
    let imgObj=refs['9cafb1a2-3b28-4468-81bb-b207501ea036']
    #文本组件
    let textObj=refs['7def40b9-61c1-4941-8637-7f46f408547c']
    if(params.value==1){
        imgObj.$el.style.display="block"
        textObj.dataChart.value='显示'
```

```
    }else{
        imgObj.$el.style.display="none"
        textObj.dataChart.value='隐藏'
    }
}
```

二次开发

> 如果系统中组件无法满足我们的应用场景，我们可以自定义开发任意组件来集成

外部引入

外部组件

> 组件文件路径/public/components.js

组件模块

> 编写组件的模块/public/components.js

```
You, 5 days ago | 1 author (You)
const testComponents = {
  template: `
  <div :style="[styleSizeName,styleName]"
    :class="className">
    <div :style="styleChartName">
      <h2>自定义组件</h2><br />
      <h3>我是参数:{{option}}</h3><br />
      <h3>data:{{dataChart}}</h3><br />
      <h3>params:{{(dataAxios.config || {}).params}}</h3><br />
    </div>
  </div>
`,
  name: 'test',
  props: {
    option: Object,
    component: Object
  },
  computed: {
    styleName () {
      return {
        fontSize: this.fontSize,
        color: this.color
      }
    },
    color () {
      return this.option.color || '#fff'
    },
    fontSize () {
      return (this.option.fontSize || 30) + 'px'
    }
  }
}
```

配置模块

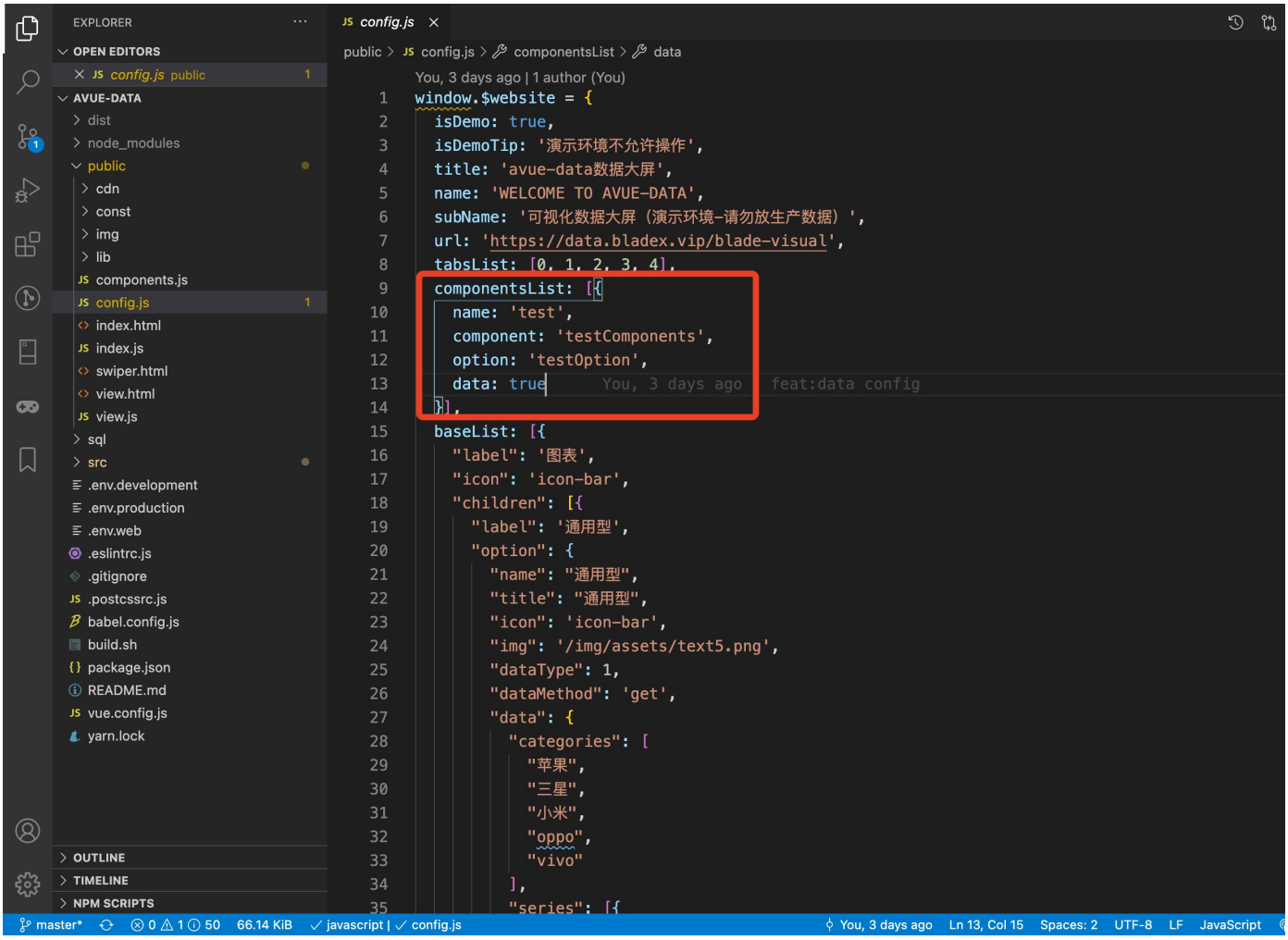
> 编写组件配置项模块/public/components.js

外部引入

```
const testOption = {
  template: `
  <div>
    <el-form-item label="字体大小">
      <avue-input-number v-model="main.activeOption.fontSize"></avue-input-number>
    </el-form-item>
    <el-form-item label="字体颜色">
      <avue-input-color v-model="main.activeOption.color"></avue-input-color>
    </el-form-item>
  </div>
`,
  name: 'test',
  inject: ["main"]
}
```

组件配置

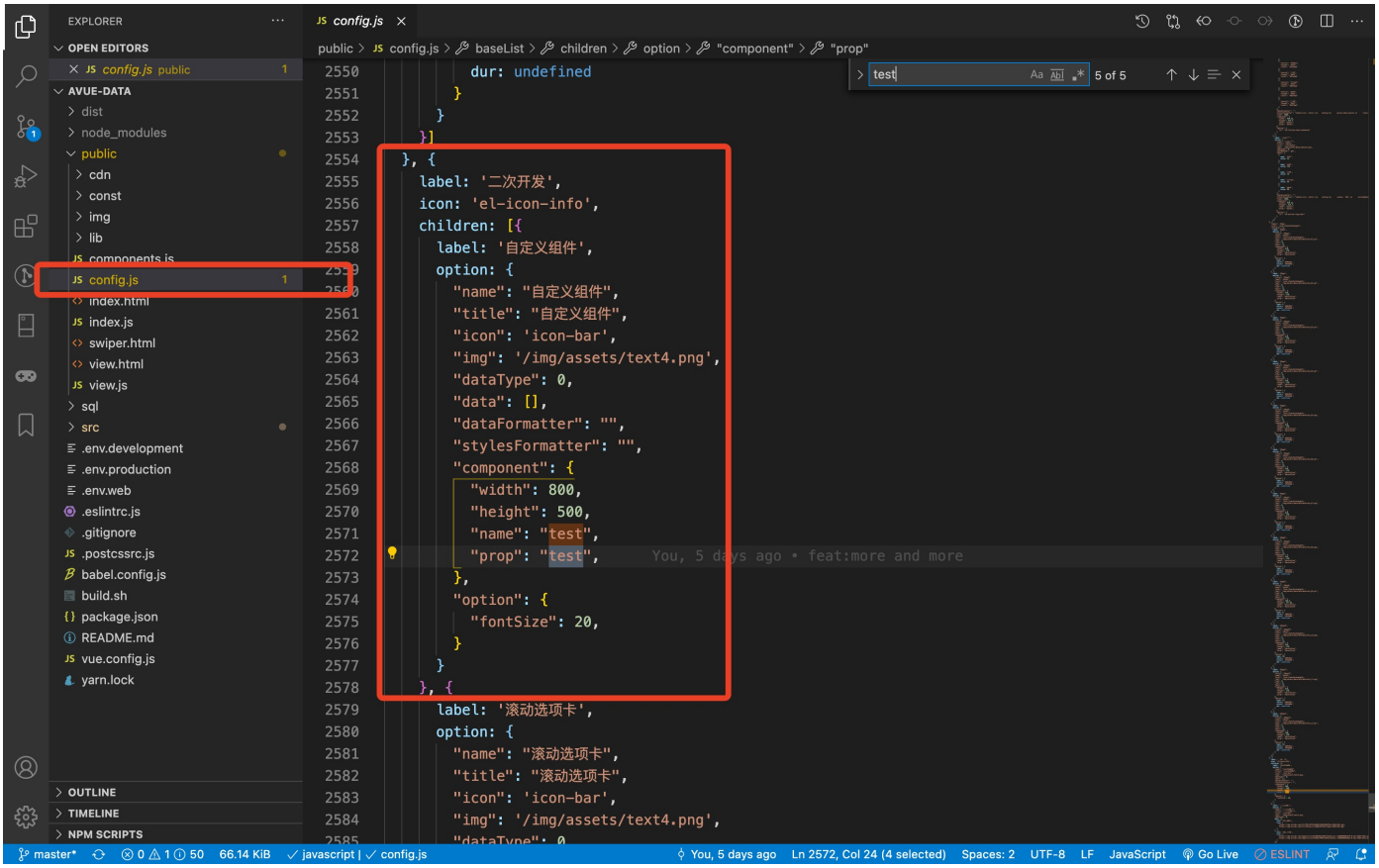
配置文件文件路径/public/config.js



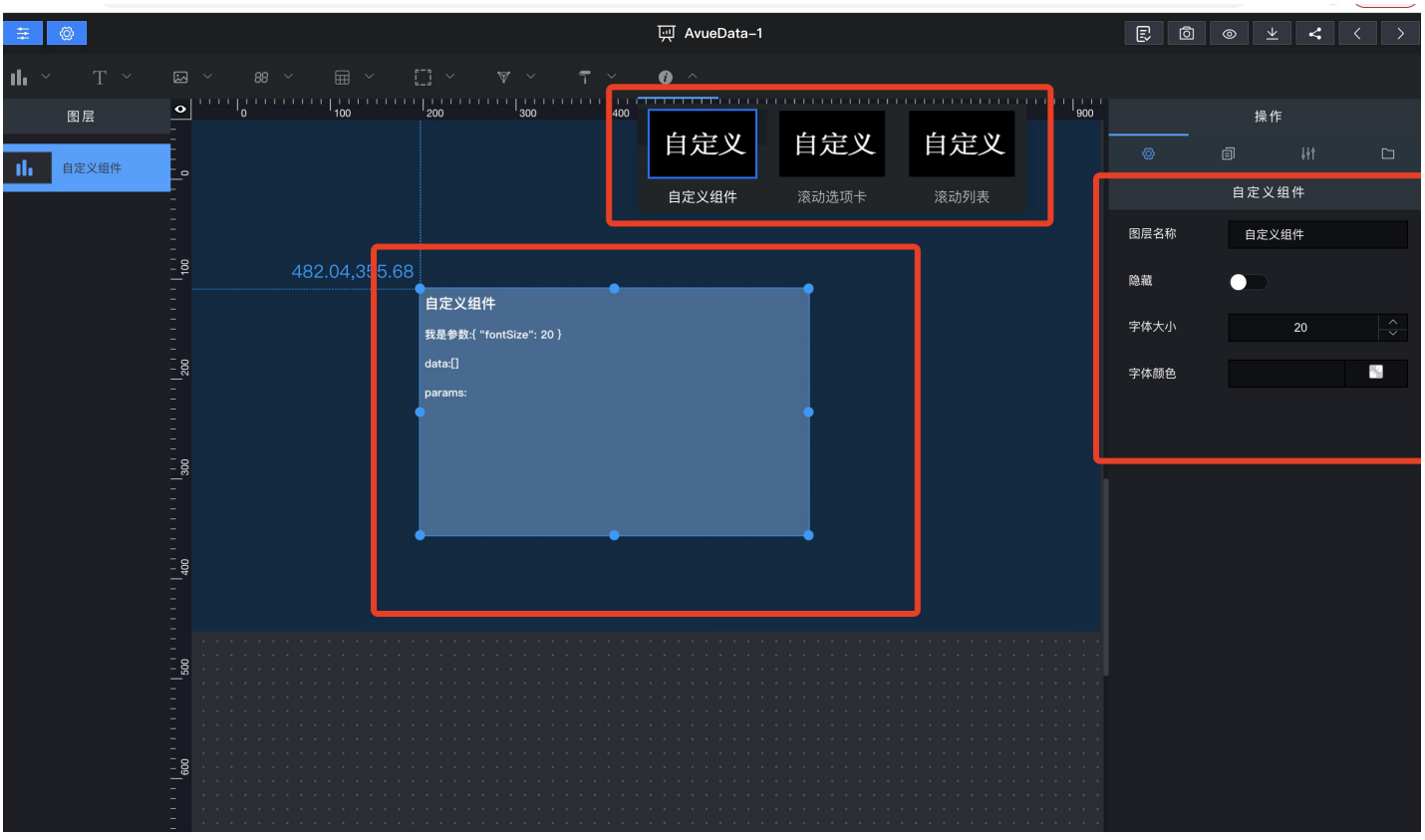
组件引用

配置文件文件路径/public/config.js

外部引入


EXPLORER
OPEN EDITORS
config.js public
AVUE-DATA
dist
node_modules
public
cdn
const
img
lib
components.js
config.js
index.html
index.js
swiper.html
view.html
view.js
sql
src
.env.development
.env.production
.env.web
.eslintrc.js
.gitignore
.postcssrc.js
babel.config.js
build.sh
package.json
README.md
vue.config.js
yarn.lock
OUTLINE
TIMELINE
NPM SCRIPTS
public > config.js > baseList > children > option > "component" > "prop"
test
5 of 5
dur: undefined
label: '二次开发',
icon: 'el-icon-info',
children: [{
label: '自定义组件',
option: {
"name": "自定义组件",
"title": "自定义组件",
"icon": 'icon-bar',
"img": '/img/assets/text4.png',
"dataType": 0,
"data": [],
"dataFormatter": "",
"stylesFormatter": "",
"component": {
"width": 800,
"height": 500,
"name": "test",
"prop": "test",
You, 5 days ago • feat:more and more
},
"option": {
"fontSize": 20,
}
}
}, {
label: '滚动选项卡',
option: {
"name": "滚动选项卡",
"title": "滚动选项卡",
"icon": 'icon-bar',
"img": '/img/assets/text4.png',
master*
0 1 50
66.14 KiB
javascript | config.js
You, 5 days ago
Ln 2572, Col 24 (4 selected)
Spaces: 2
UTF-8
LF
JavaScript
Go Live
ESLINT

最终效果


AvueData-1
图层
自定义组件
自定义
自定义
自定义
自定义组件
滚动选项卡
滚动列表
操作
自定义组件
图层名称
自定义组件
隐藏
字体大小
20
字体颜色
自定义组件
我是参数:{ "fontSize": 20 }
data:[]
params:

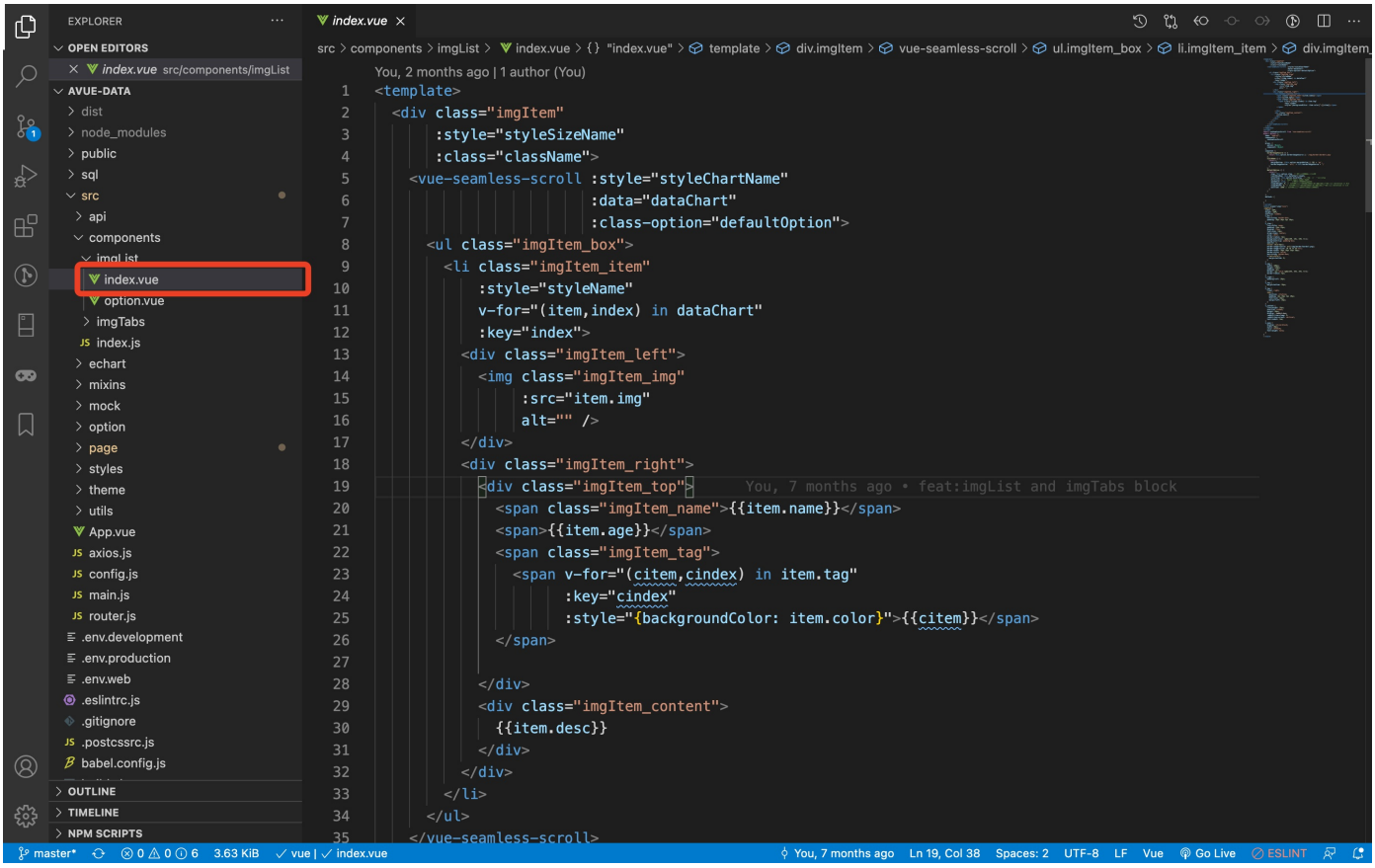
内部引入

内部组件

> 组件文件路径/src/components/**.js，配置对应组件，系统会自动引入模块

组件模块

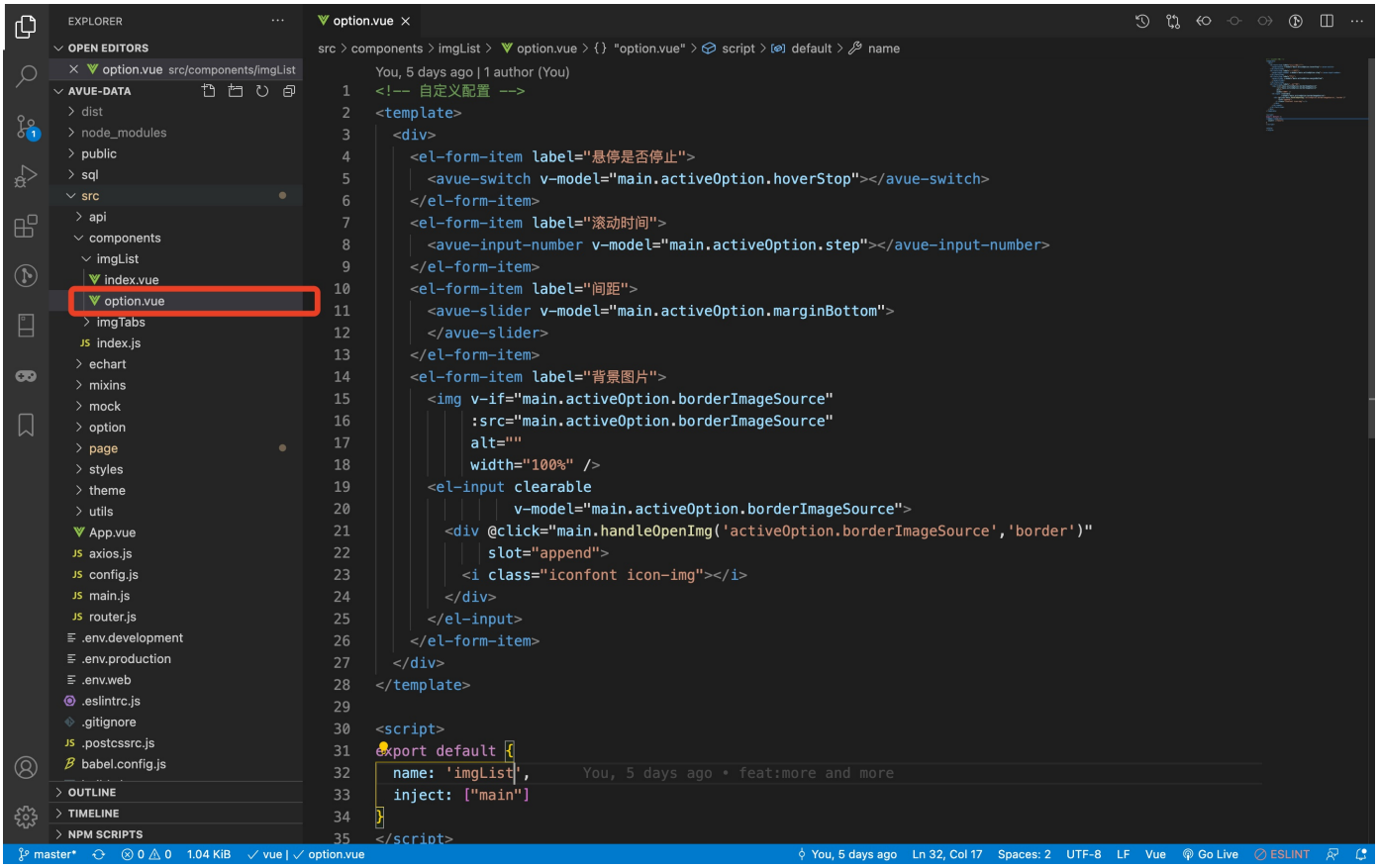
> 编写组件的模块/src/components/imgList/index.vue



配置模块

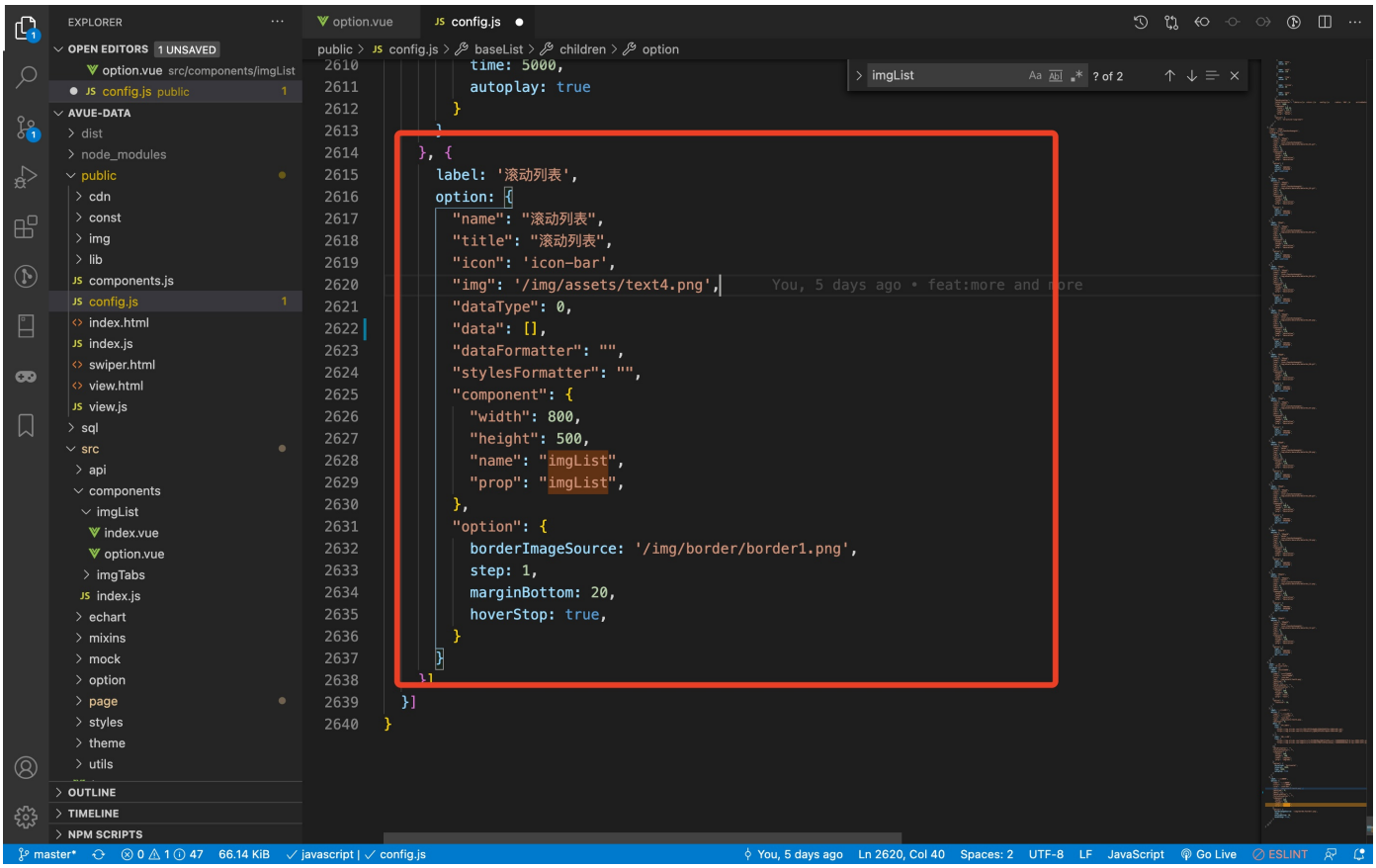
> 编写配置模块/src/components/imgList/option.vue

内部引入

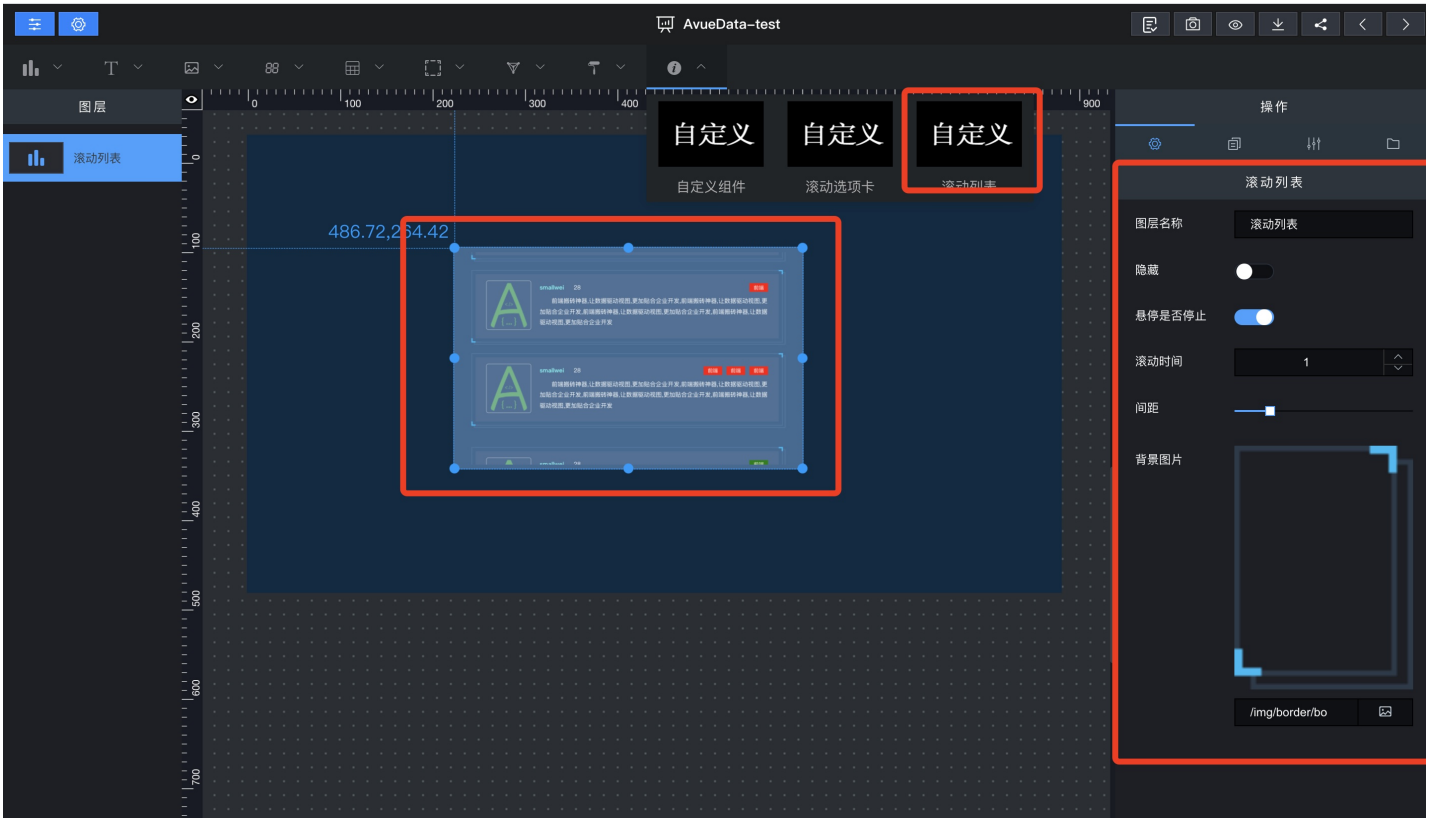


组件引用

配置文件文件路径/public/config.js



最终效果


AvueData-test
图层
滚动列表
自定义
自定义
自定义
自定义组件
滚动选项卡
操作
滚动列表
图层名称
滚动列表
隐藏
悬停是否停止
滚动时间
1
间距
背景图片
/img/border/bo

项目部署

Nginx部署
1.项目打包
2.Linux部署
3.Baota部署

Nginx部署

> 执行npm run build打出对应的目录

```
//nginx为例子
location /{
    root /data/avue/avue-data;  //打包出的路径
    index index.html;
    error_page 404 /index.html; //根据vue-router路由history特性，这句一定要配置，否则会出现404问题
}
```

1.项目打包

一、后端打包

1. 将相关配置放到对应的文件

2. 根目录执行 `mvn clean package -Dmaven.test.skip=true`

3. 打包成功后我们便可以看到blade-visual.jar这个jar包，拷贝出来备用

二、前端打包

1. 配置好服务器ip与端口到对应的接口url

2. 根目录执行 `yarn run build`

3. 打包成功后，我们便可以看到 `dist` 文件夹下的目录，拷贝出来备用

2.Linux部署

一、服务器准备

1. 服务器上部署好mysql、redis、minio、java，此处不再赘述

二、后端部署

1. 将上一章节打包好的jar包上传，注意要提前将服务器的配置加上
2. 执行命令 java -jar blade-visual.jar 启动

3. 后端启动成功后访问对应地址，若显示如下界面则说明后端启动成功

三、前端部署

1. 将上一章节打包好的dist上传，注意要提前配置好后端api地址

2. 到nginx配置文件进行配置增加如下配置，否则会出现404

```
location /{
    root /usr/share/nginx/html;
    index index.html;
    error_page 404 /index.html;
}
```

3. 打开服务器地址，查看结果，可以正常访问，说明部署成功

3.Baota部署

3.1安装宝塔

3.2基础部署

3.3大屏部署

3.1安装宝塔

一、开始安装

> 注意⚠：宝塔部署需要一个纯净的操作系统，切勿在已有环境尤其是生产服务器上安装！

1. 登录已经准备好纯净系统的服务器, 请自行选择熟悉的软件连接登录
2. 执行如下命令自动安装

```
[root@blade-test ~]# yum install -y wget && wget -O install.sh http://downl
oad.bt.cn/install/install_6.0.sh && sh install.sh
```



3. 注意记录最底部的面板地址、username、password，不要丢失，然后访问地址输入帐号密码进行登录

4. 初次登陆也许会相对耗时，此时系统在初始化，稍等片刻就可以看到进入了面板欢迎页，选择同意并进入面板

5. 进入首页后会弹出默认环境安装，我们选择左侧的一键套件，选择mysql5.7，点击一键安装后耐心等待安装完毕


6. 绑定宝塔官方的帐号，若未注册可访问这个链接：https://www.bt.cn/?invite_code=MV9namxtdXM=

7. 完整首页，看到这一步就说明可以了

二、创建站点

1. 宝塔基础环境安装完毕，下面我们来尝试创建一个站点，用于给后续的前端部署做准备
2. 我们先创建一个站点，格式为外网ip+端口，注意到云服务器的安全组规则给端口开放访问权限


3. 访问设定的ip和端口，可以看到显示创建成功

4. 点进根目录，找到index.html，修改一下内容


5. 刷新界面，发现刚添加的信息已经显示成功，说明站点创建成功

三、部署域名

- 很多时候项目的正式环境都是用于域名访问
- 使用域名的同时也会使用https的形式
- 下面我们演示如何部署一个域名并且配置免费https证书

四、新建站点

1. 给站点起名使用域名形式，这里我们写为 `web.bladex.vip` 并且给域名做好映射


2. 稍等片刻，访问http://web.bladex.vip，可以看到域名访问生效

五、配置Https

1. 打开SSL栏目，点击免费的Let's Encrypt，选中域名并点击申请



2. 再次刷新https://web.bladex.vip，可以正确访问，并且可以看到证书信息


3.2基础部署

一、安装Java环境

1. 前往软件商店搜索 `java` 点击安装，并首页显示

2. 点击设置，选择安装tomcat8服务，这里主要是为了安装java8的环境

二、安装Redis环境

1. 前往软件商店搜索 `redis` 点击安装，并首页显示，若是生产环境，建议修改端口以及密码，并且不开放至外网访问

三、配置Mysql

1. 创建bladex数据库，导入脚本，并将帐号密码记录好后续放到 `blade-visual` 的 `application-test.yml` 配置文件

2. 点击管理，查看数据是否导入成功

注意

- java、redis、mysql三个基础环境是必须要的，其中尤其是redis与mysql，务必确认设置密码并且日常关闭外网访问
- 若都采用默认配置，不出意外，服务器很快就会被人挂上挖矿病毒，这个请再三确认，不希望大家的服务器因此挂掉

3.3大屏部署

一、站点创建

1. 前往宝塔新建一个web站点

2. 站点访问成功

二、后端部署

1. 将部署好的配置放入application-test.yml

2. 根目录执行 `mvn clean package -Dmaven.test.skip=true`

3. 打包成功后我们便可以看到blade-visual.jar这个jar包，准备上传

4. 点击进入站点目录，并创建api文件夹

5. 将blade-visual.jar上传至api文件夹

6. 将BladeX的启动脚本稍作修改也上传至api文件夹并配置好权限

> 默认端口为8050，若我们需要改成其他端口，比如88，给启动命令加上 `--server.port=88`
> 为了生效 `application-test.yml` ，给启动命令加上 `--spring.profiles.active=test`

三、后端运行

1. 前往首页点击ssh终端并输入服务器密码

2. 进入目录运行脚本

3. 开放端口（云服务器安全组策略以及宝塔的安全端口都需要开放）

4. 稍等片刻，等待服务启动完毕，我们访问对应地址,看到如下界面则说明后端部署成功

四、前端部署

1. 前端配置刚刚部署好的后端地址

2. 根目录执行 `yarn run build`

3. 打包成功后，我们便可以看到 `dist` 文件夹下的目录，拷贝出来准备上传

4. 将目录打成zip包，并上传至刚刚创建的站点根目录并右键解压

5. 返回站点列表，进行nginx伪静态配置，否则会出现404

6. 加入配置并保存

```
location /{
    index index.html;
    error_page 404 /index.html;
}
```

五、前端运行

1. 刷新站点网站，可以看到系统已经部署成功

2. 打开f12，可以看到接口地址也正确无误

引入其它项目使用

> 选择对应的大屏模块，点就导出，将导出的文件引入其他项目即可