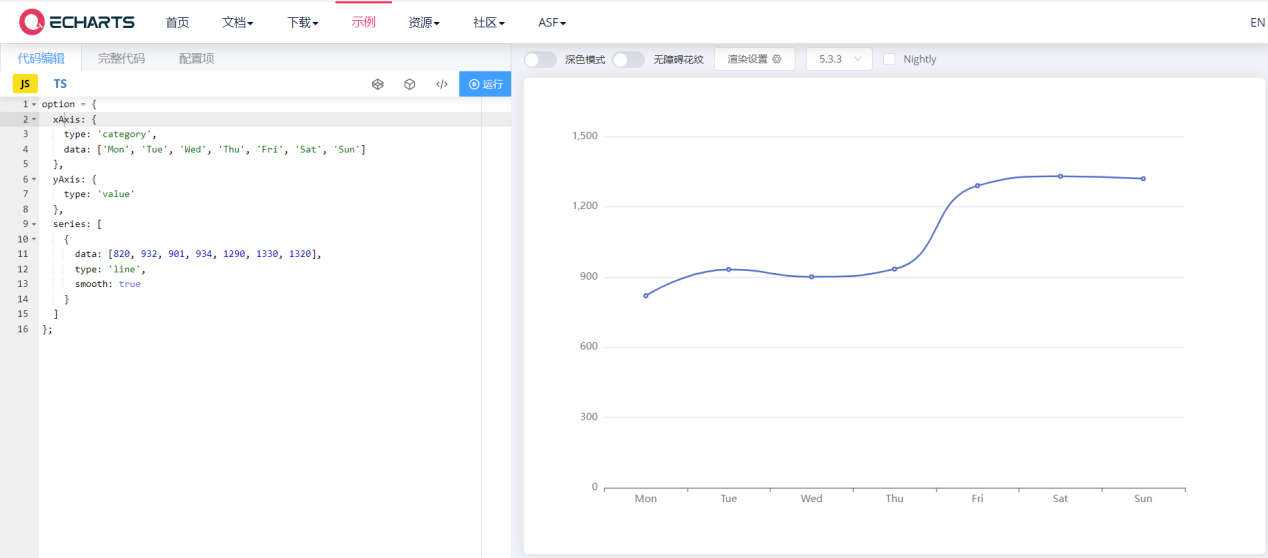
### 基于 vue 的 Apache Echarts 可视化统计图配置工具

#### （项目书）

**设计背景：**

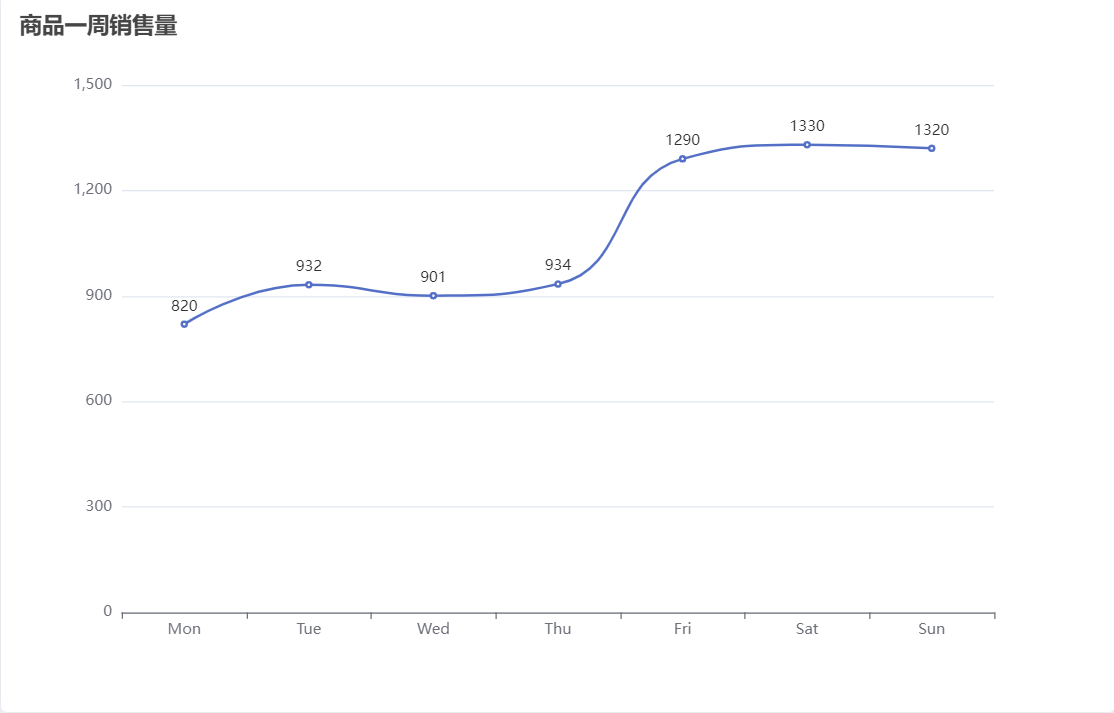
场景一：某用户想使用 Apache Echarts 统计图绘制一副有 “标题”、“数据”的基础平滑折线图。



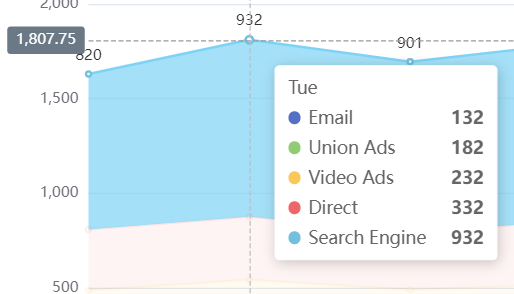
根据Apache Echarts 默认的平滑折线图，是没有标题、数据选项，因此，用户需要查看Apache提供的配置项手册：



但是！用户需要一个一个点击、测试，才能知道哪一项的配置项才是自己想要的效果。最终，用户实现如下效果：



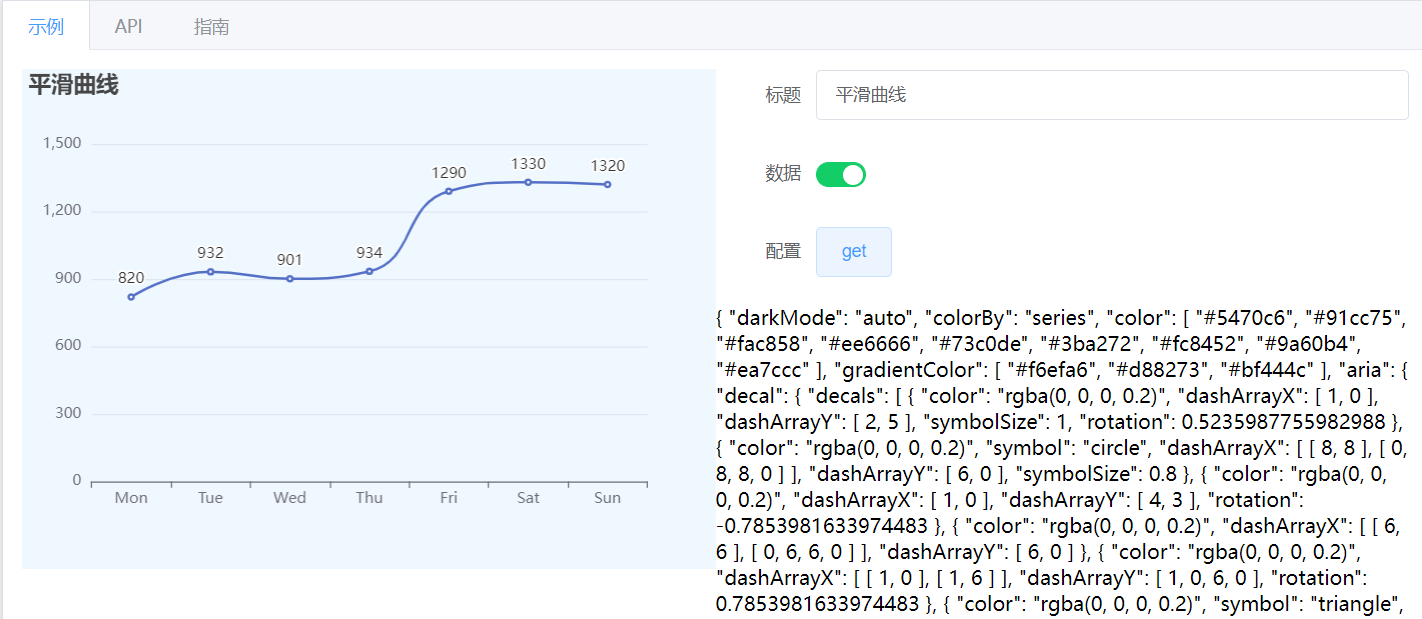
这只是简单的一个需求，再复杂的，例如：用户希望放上去能显示数据等复杂需求就难以用同等方式实现。



如果每一项用户想要的效果都需要自己查找配置项手册配置，过于繁琐、复杂，不利于用户的体验。

因此，基于 vue 的Apache Echarts 的可视化统计图配置工具营运而生：还是上某用户的需求，我们通过配置面板的按钮、输入框，实现用户想要的效果，并能通过配置项的复制功能，将生成的配置项复制到自己的项目中，无需查阅Apache的配置项手册，减少了用户的使用难度，同时，对统计图的使用成本也大大降低。

效果图如下：



后期将对配置项的获取做进一步优化，使得更符合实际需求。总而言之，配置工具使得 Apache Echarts 的使用更加简单快速。

**技术说明：**

本项目开发应用到的技术如下Vue2 + element UI + Apache Echarts：

Vue2 作为项目的框架结构，是前端页面的主要设计语言；

Element UI 是搭配vue2 使用的前端UI组件库，更加便于开发人员搭配出简介好看的界面UI；

Apache Echarts：是项目的中心，配置项的设计、可转化为配置的关键结构，需要开发人员深入了解，并实现配置式实时更新统计图，能通过配置项获取，得到简介的配置信息，并供用户复制使用。

**设计规范：**

1. 获取DOM的方式：

因使用Vue2 作为项目技术栈，因此，获取页面DOM需要符合Vue2的设计规范， 使用 ref 标记页面元素，通过 this.$refs[“refName”]的方式获取。如下示例：

标记：<div ref=”demo1”></div>

获取：var chartDom =this.$refs["demo1"];

1. 命名方式：

方法命名：使用驼峰式（Function getUserInfo()）、

组件命名：使用小写、组合时使用kebab-case (短横线分隔命名)(user-info)

引用命名：引用使用首字母大写形式命名（import GetUser from xxx）

其他命名参见[W3C规范](https://www.w3cschool.cn/wematy/un29iozt.html)。

1. 方法注释

方法（methods）：必须写注释，并注明方法作用，参数意义等

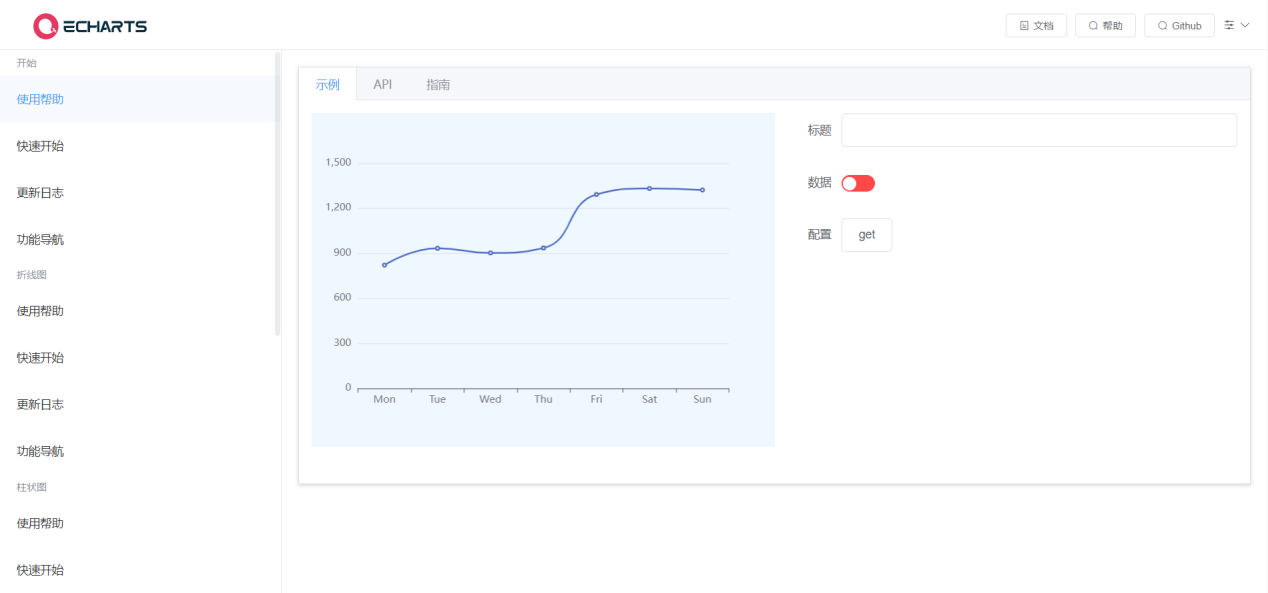
页面结构：复杂的页面结构需写注释，包括但不限于页面深层次嵌套、复杂的布局

样式（CSS）：css中不常用的样式需要写清注释，对页面布局有关键意义的需注释

**协同开发：**

代码托管GitHub：<https://github.com/wf0/vue-Apache-Echarts-options>

**功能设计：**



项目的框架已经搭建完成，后期的主要工作是：

1. 开发相应的统计图配置项；
2. 逐步完善项目；
3. 优化页面UI；