# 可视化技术与 Vue 介绍

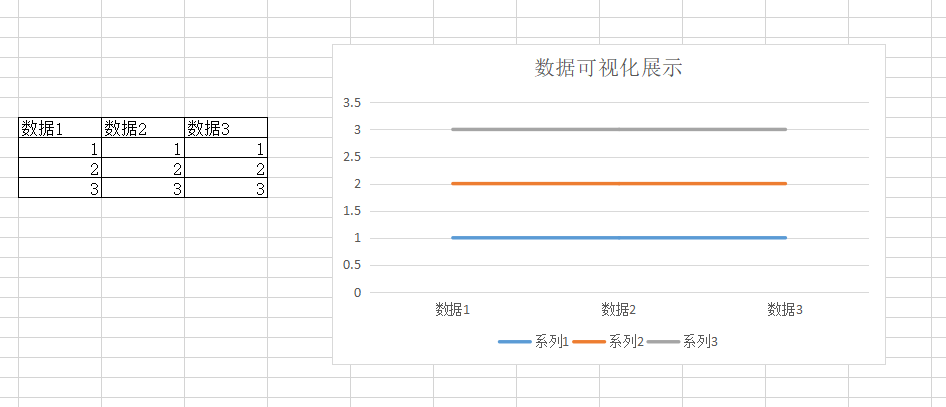
## 什么是可视化

“数据可视化” 是一门数据与视觉相结合的技术，简单理解就是将数据转换成易于用户辨识和理解的视觉表现形式。通过图表展现方式，让数据转换为视觉图像，如各种 2D 图表、3D 图表、地理地图、关系数据图等等，以此来更好的展现数据的价值。

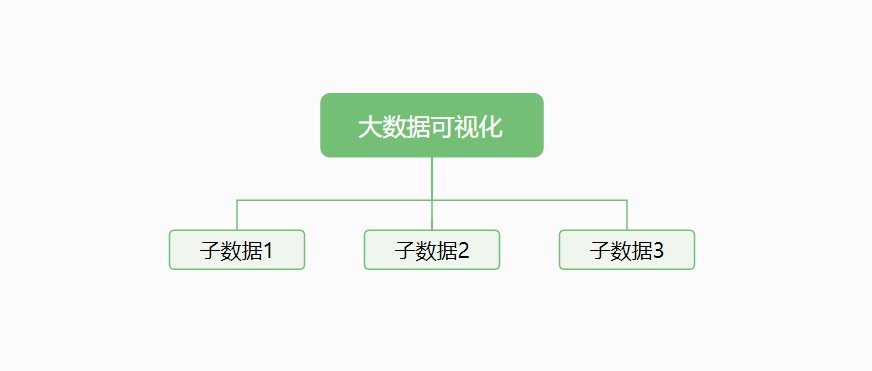
#### 可视化应用场景

在此我们举几个十分常见的可视化简单应用，来加深大家的印象。

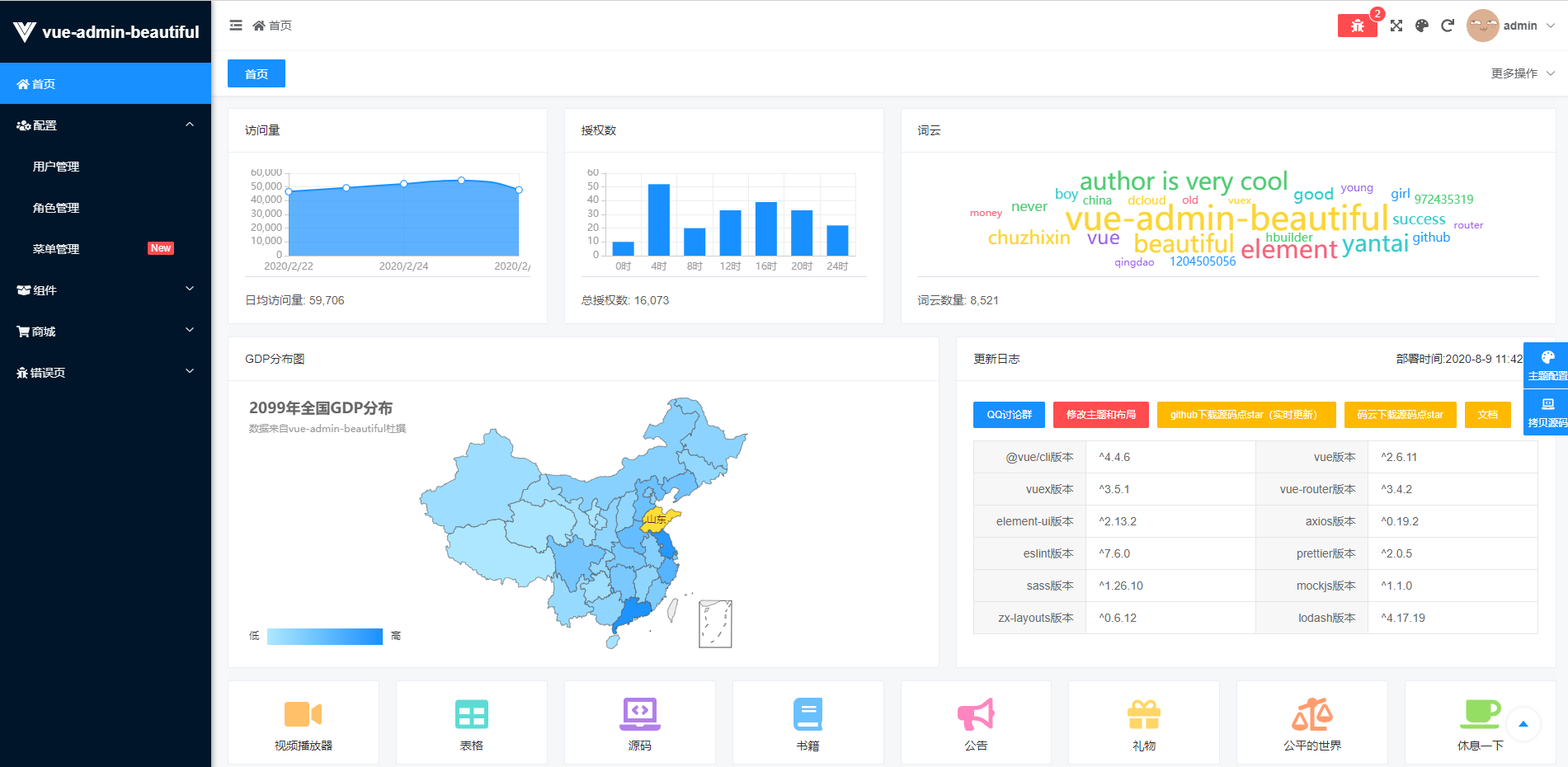
* Excel



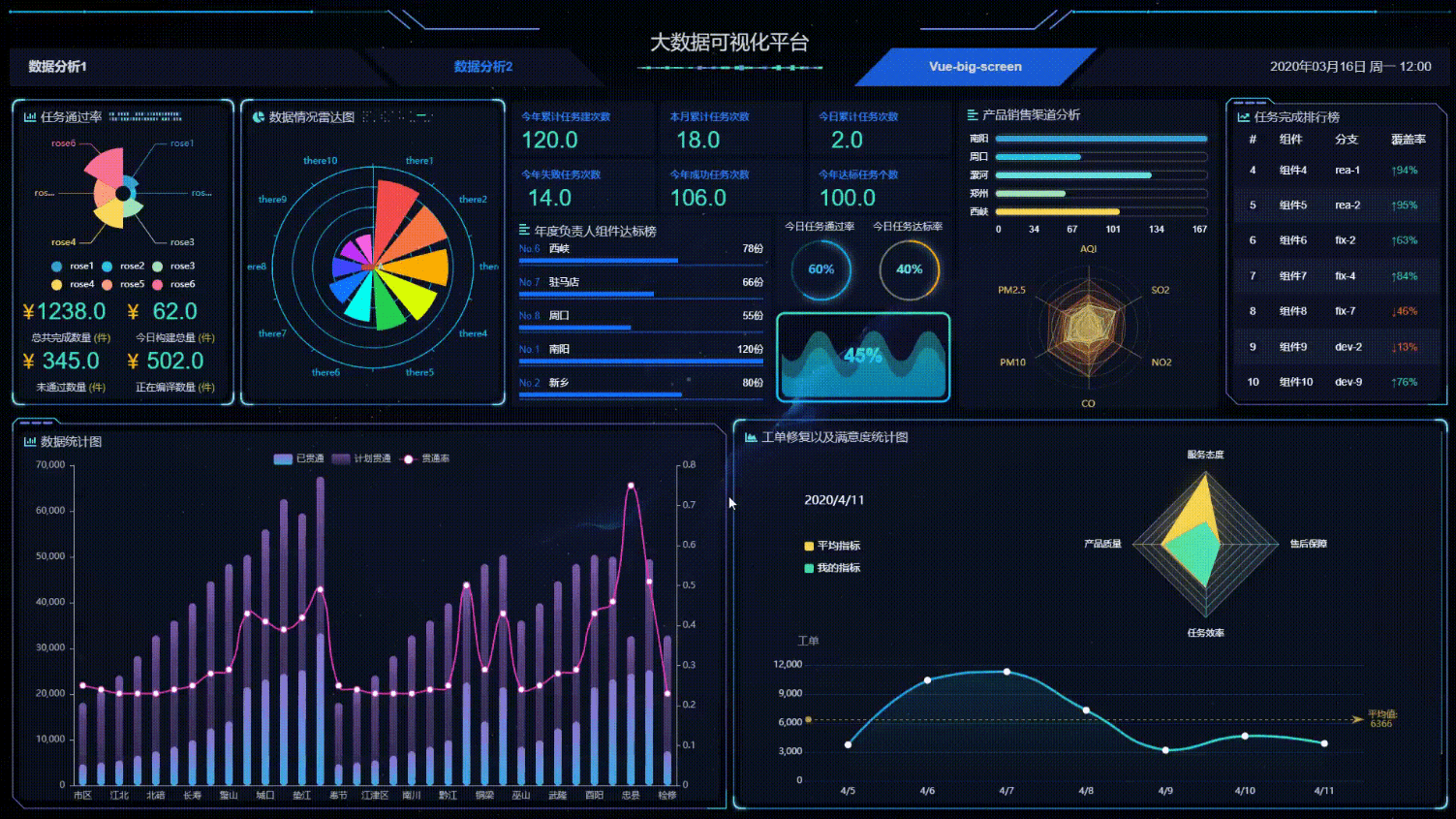
* 百度脑图



* 项目数据展示（vue-admin-beautiful）



* 数据大屏



## 可视化技术的框架与优缺点

#### 可视化技术框架介绍



可视化技术运用广泛，但是实现起来难度较大，使用原生写法门槛较高，于是在技术社区的氛围之下也诞生出了非常多优秀的开源框架，这里也列举了一些经典框架并相应地介绍它们的优缺点：

| **框架** | **说明** |
| --- | --- |
| Echart.js | 国内最火的图表库之一，由百度技术团队出品，内容丰富社区活跃，Echarts3 对性能进行了很好的提升和改进。 |
| AntV | 蚂蚁可视化技术，类别丰富，拥有 G2、G6、F2、L7 等专业图标库，对图形的控制上要更自由，更抽象通用。 |
| Highcharts | 内容丰富，支持 IE6 +、iPhone/iPad、Android 等目前市面上所有主流的浏览器，即开创性的支持 IE6 浏览器。 |
| D3.js | D3 是最流行的可视化库之一，它被很多其他的表格插件所使用。功能拆得非常细，适合拿来学习的可视化算法。 |
| Chart.js | 一个简单、面向对象、为设计者和开发者准备的图表绘制工具库。 |

## 可视化技术的优缺点

#### 优点

优秀的可视化图形可以帮助人们快速、轻松地理解数据之间的联系，其显著的特点是能够让用户通过快速地吸收图形特征，从而理解信息内容，达到高效率提升数据分析的目的。

可视化技术在很大程度上减轻了用户对于数据表的理解压力，无需让用户埋头于大量令人眼花缭乱的数据中。可视化技术也广泛地应用在大屏展示相关场合，通过本身自带的良好视觉效果属性极大的提升了推广效率。

#### 缺点

可视化技术的原生数据交互能力拥有极大的局限性，并且展示效果也会受到图像风格的限制。由于本身的特性导致其能展示的报表级别不会太高，无法进行更进一步的展示。如果不能合理的将业务数据与图表形式进行结合，则效果可能会不尽如人意，违背了初衷。

## Vue 介绍

#### Vue 技术

Vue 是一套用于构建用户界面、实现 MVVM 开发方式的渐进式框架。根据官网的介绍说明，可以看出与其它传统 JQuery、Bootstrap 等大型框架不同的是，Vue 的核心在于只关注视图层的同时，避免了直接操作 Dom 元素，让开发者更好地处理数据直接的联系，并且能够为复杂的单页应用提供驱动。

通常在项目的应用中开发者更多的是对 Vue 与其技术体系插件进行使用，包括但不限于：Vue-router、VueX、Vue-cli、Axios、Element-ui 等。

与 Vue 对应的还有大家熟悉的 React 和 Angular 框架，相比于 Vue 来说，Angular 和 React 的学习成本都很高，对新人不太友好，并且中国使用 Vue 的公司是占大多数，所以为了让大家更快地上手和在实际中进行应用，本课程选择使用 Vue 技术进行本课程开发实现。

#### Vue-cli 脚手架

Vue-cli 是一个基于 Vue.js 进行快速开发的完整系统，也称之为 Vue 脚手架。它只需通过便捷 npm/yarn 进行安装即可，通过几行简单的命令，就能够帮助开发者搭建一个完整的项目框架，节省大量前期开发配置时间。更多详细介绍说明可以在 [官网](https://cli.vuejs.org/zh/guide/) 进行查看。

官方说明它具备以下几点主要特征：

* 可升级、基于 Webpack 构建，并带有合理的默认配置。
* 可以通过项目内的配置文件进行配置。
* 可以通过插件进行扩展。
* 一个丰富的官方插件集合，集成了前端生态中最好的工具。
* 一套完全图形化的创建和管理 Vue.js 项目的用户界面。

Vue-cli 还具备界面 UI 工具，开发者可以用来创建和管理项目，简化了编写命令行的步骤。开发者可以在浏览器中配置或运行 Vue 项目相关的所有任务，包括依赖项、插件、命令等。项目的可视化演示使此工具，能够让开发者直观快速的进行使用，而在本教学课程中，项目和初始练习项目的环境也都是选择了脚手架来进行搭建。

# 项目搭建与 Vue 基础讲解

## 借助 Vue-cli 搭建项目

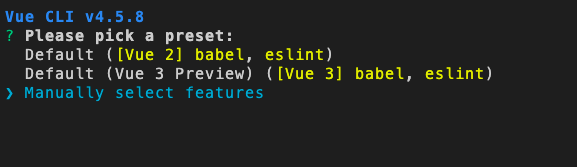
实验楼环境中已经为我们安装好了 Vue-cli，我们可以输入命令 vue --version 查看版本信息：

vue --version

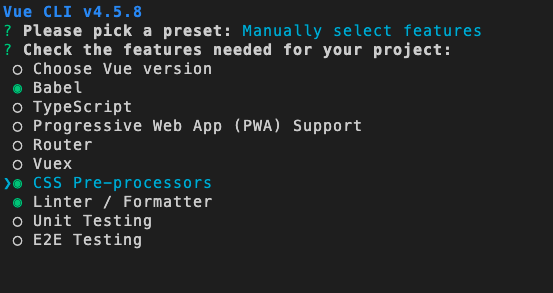
我们输入命令创建一个名为 vue-mt-system 的项目：

vue create vue-mt-system

此时会提示我们进行选择，由于后面的课程会进行 Sass 的教学，所以我们选择末尾的自定义选项（脚手架的默认配置中是没有 Sass 依赖的）：



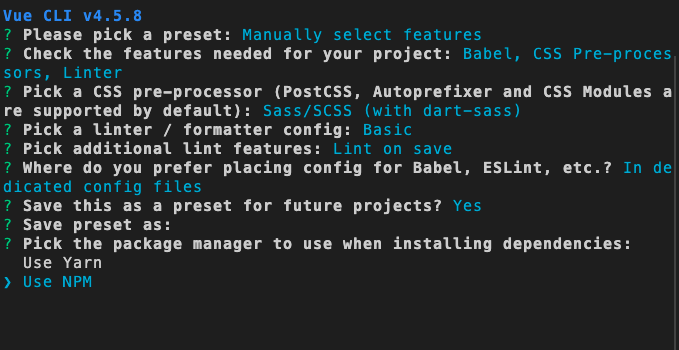
我们需要使用**空格键**来确认 CSS Pre-processors 选项，按下回车键进行确认。



后面的配置（除了最后一项）都是选择默认项所以 **直接回车** 即可。

Save this as a preset for future projects? 输入 y。

最后一项是选择安装依赖的方式，我们选择 Use NPM。



创建完毕之后我们通过命令行进入目标文件夹。

cd vue-mt-system

为了满足实验楼的 Web 服务要求，我们需要进行一些配置。我们在 vue-mt-system 目录下创建名称为 vue.config.js 的配置文件（文件名称要一致），在里面填入下面内容：

const HOST = process.env.HOST;

module.exports = {

publicPath: './',

productionSourceMap: false,

devServer: {

host: HOST || '0.0.0.0',

open: true,

disableHostCheck: true,

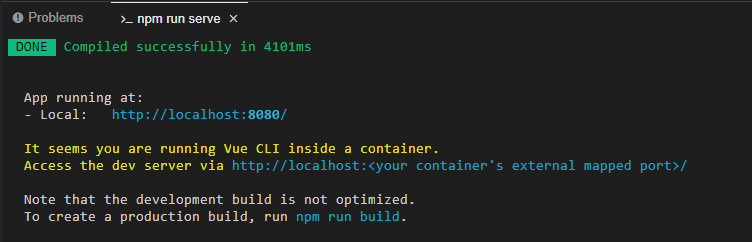
},

};

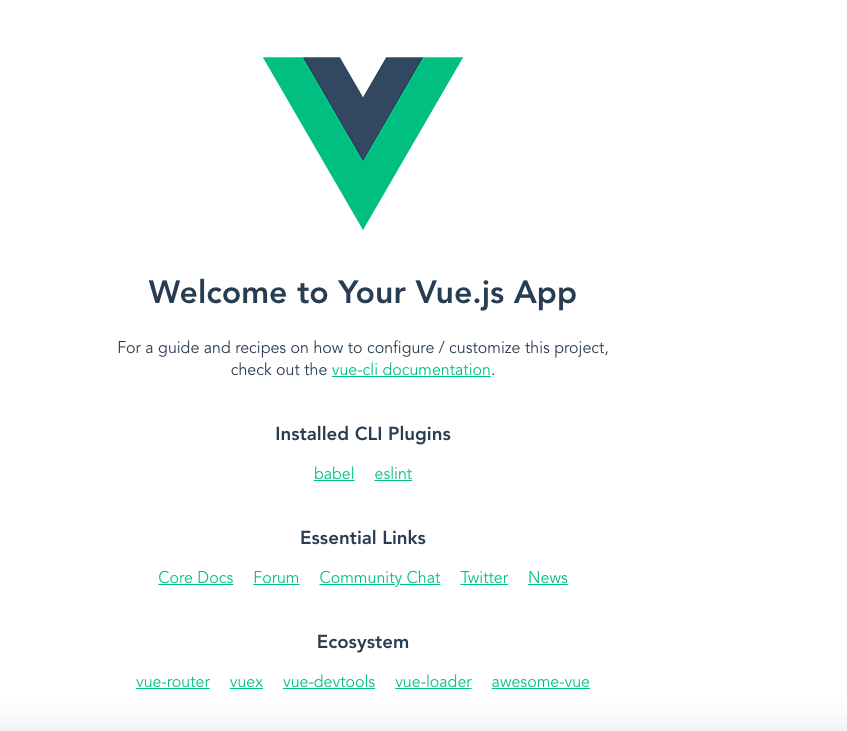
写好配置文件后，使用以下命令启动项目：

npm run serve

出现以下结果就说明项目已经编译成功了：



我们点击实验楼右侧选择列表中的 **Web 服务** 按钮，就可以打开界面并将看到 Vue-cli 创建的初始效果。

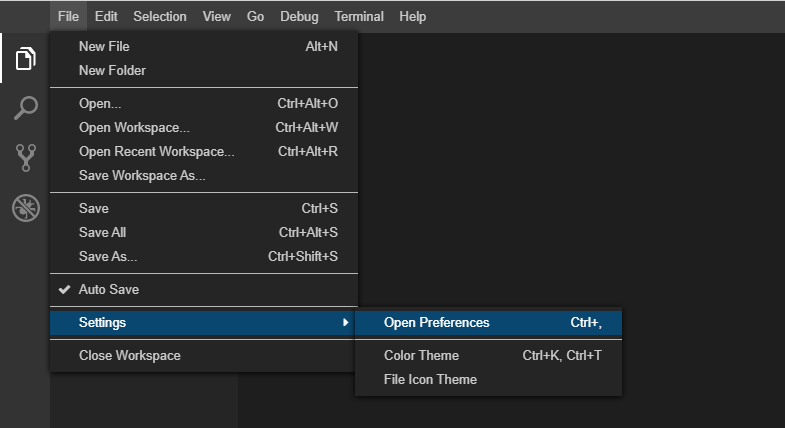


关于 Vue-cli 项目目录功能分类以及其它细节讲解，我们会放到实战的时候细说，前期主要目的是进行基础技能的学习。

#### 修改 VsCode 配置

由于实验楼的 WebIDE 具有自动保存的功能，但是保存的时间间隔太短了，以至于项目一直在不断地进行热更新，容易造成卡顿，我们可以进行适当的修改。

点击 File -> Settings -> Open Preferences 进入配置界面：



点击左侧的 Editor ，找到 Auto Save 选项，将下方的 Delay 改成 2000（2 秒），如果不想要自动保存功能也可以直接选择 off 进行关闭，编写代码之后按 ctrl + s 即可。

