# Vue 3 介绍与安装

## Vue.js 介绍

Vue 是一套用于构建用户界面的渐进式框架。与其它大型框架不同的是，Vue 被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合。另一方面，当与现代化的工具链以及各种支持类库结合使用时，Vue 也完全能够为复杂的单页应用提供驱动。<信息来源：vue 官网>

Vue 是一个推陈出新的前端框架，吸收了很多前端框架的优点。例如，Vue 借鉴了 React 的组件化和虚拟 DOM ，借鉴了 Angular 的模块化和数据绑定。因此，我们以 Vue.js 作为入手，以后深入学习其他框架，你会发现他们的共通之处，更好地高效地学习。

选择 Vue.js 的更多原因是，就框架的 API 而言，对比之下，Vue 更加轻便简洁。Vue 自身拥有开箱即用的生态开发包（Vuex,Vue-Router）等，复杂性低、学习成本低，是一门比较好入门的前端框架。

## Vue 3 的特性

Vue 3 相对与之前的版本，有 6 个方面的重要变更：

* Performance （性能）。
* Tree-shaking support （支持 Tree-shaking）。
* Composition API。
* Fragment, Teleport, Suspense。
* Better TypeScript support （更好的 TypeScript 支持度）。
* Custom Renderer API （自定义的 Renderer API）。

## script 标签引入安装

标签引入模式，和平常引入 JQuery 一样，直接下载并用 <script> 标签引入，我们使用的是最新的 CDN：

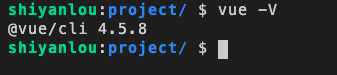
<script src="https://labfile.oss.aliyuncs.com/courses/2495/vue.global.js"></script>

## 脚手架工具安装

在本实验中我们使用脚手架工具 vue-cli 的方式来初始化项目。使用以下命令来安装脚手架工具：

npm i -g @vue/cli

安装完成后，我们用 vue -V 可以看到目前的版本。

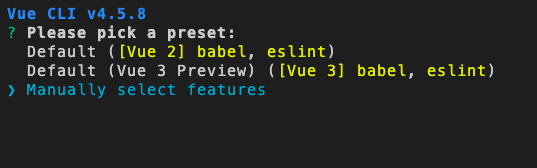


然后我们初始化一个名为 vue-next-test 的 Vue 项目。

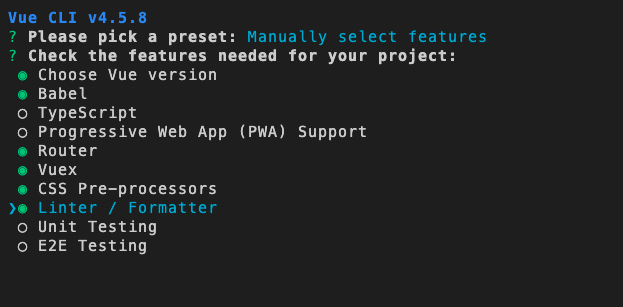
vue create vue-next-test

输入命令后，会出现命令行的交互窗口，交互窗口的命令较多，希望同学们可以跟着实验步骤来选择。

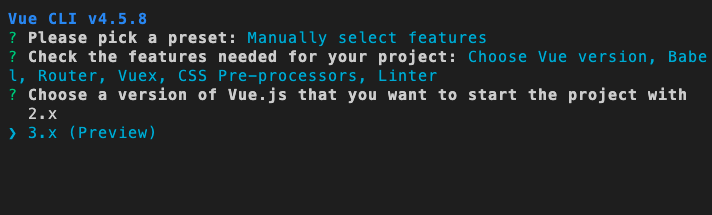
如下所示第一个选项，我们选择 Manually select features：



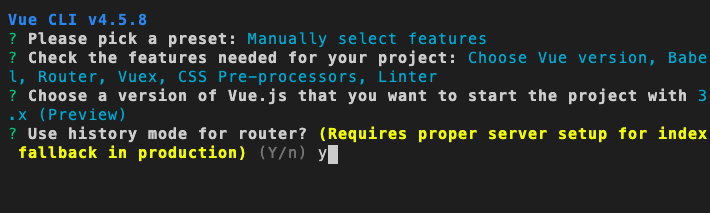
选好之后点击回车。在第二个选项中，我们勾选 Router、Vuex、CSS Pre-processors 和 Linter / Formatter，移动到选项后点击空格即可勾选，勾选完成后敲入回车进入下一步：



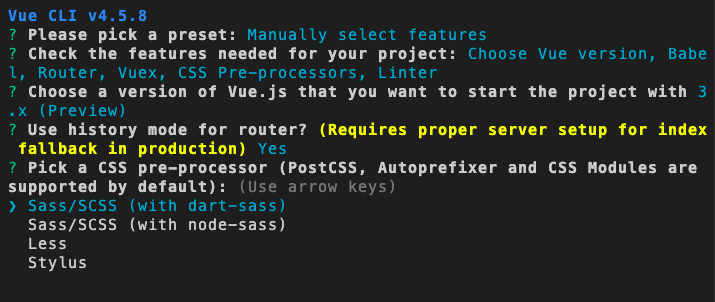
在第三个选项中，我们选择 3.x（Preview）：

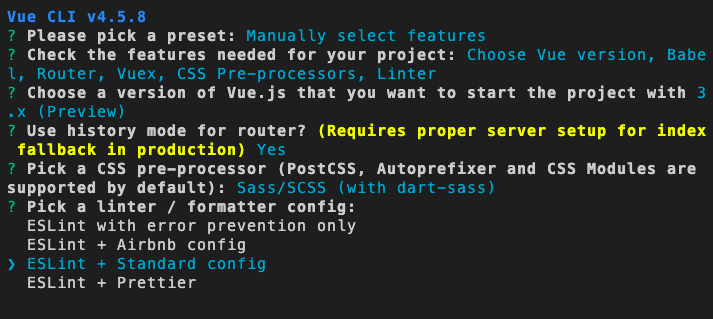


选择使用 history 路由模式，即输入 y：

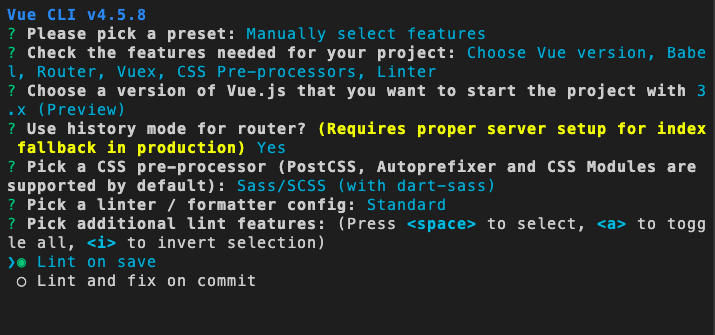


接下来选择 css 预处理器，选择第一个即可 Sass/SCSS，下一步选择 eslint 配置：ESLint + Standard config：

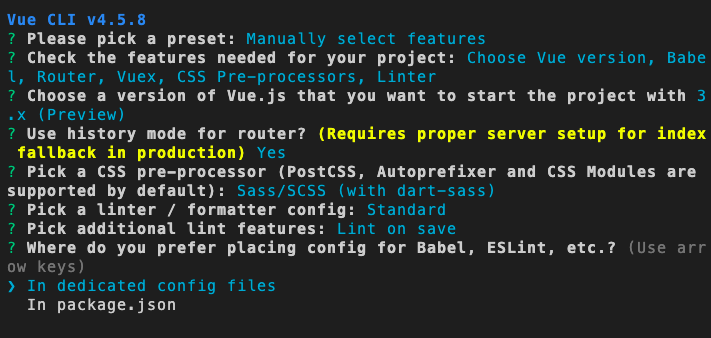




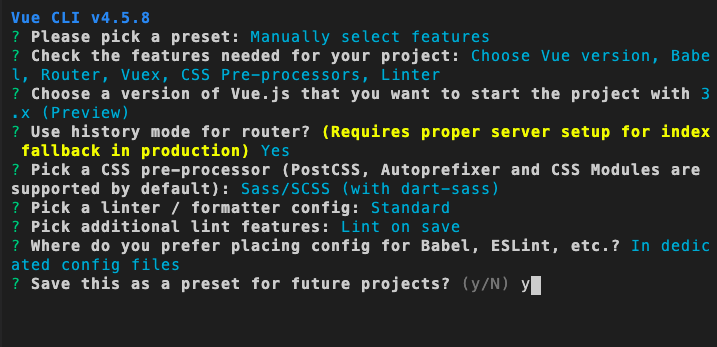
选择什么时候执行 eslint 校验，我们选择 Lint on save：



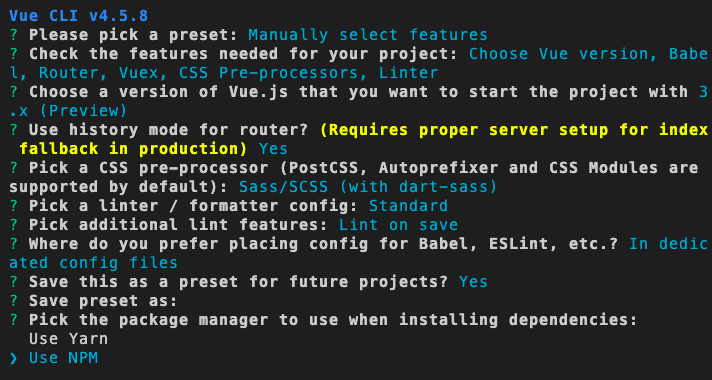
接下来是选择以什么样的形式配置以上所选的功能，选择第一个 In dedicated config files：



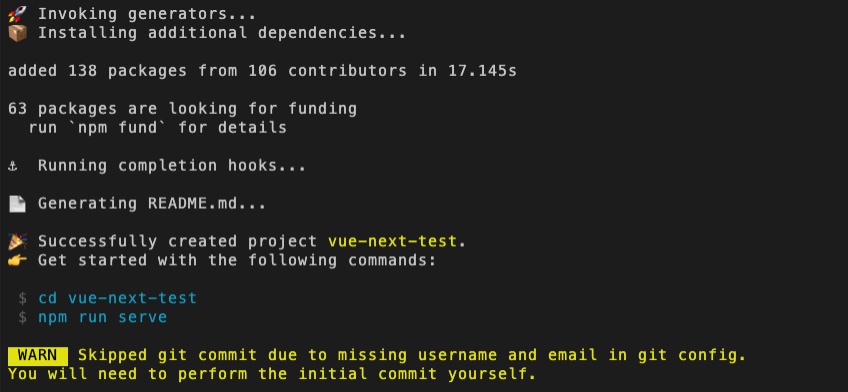
下面一个选项是是否将之前的设置保存为一个预设模板，我们输入 y：



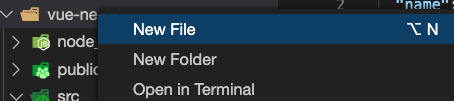
对于下面所出现的 Save preset as 直接回车即可。接下来我们选择 npm 来安装依赖：

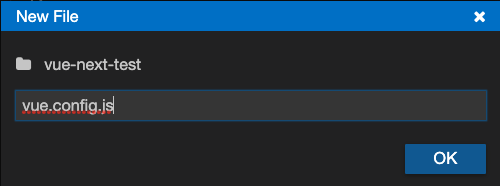


敲入回车后，进入安装模式，如果看到以下截图的内容，说明安装完成。



这样我们就创建了一个 Vue 3 的项目。由于我们的线上环境只开放 8080 端口，所以我们需要的 vue-next-test 目录下创建一个 vue.config.js 的文件。





在 vue.config.js 文件中写入以下内容：

module.exports = {

devServer: {

disableHostCheck: true,

},

};

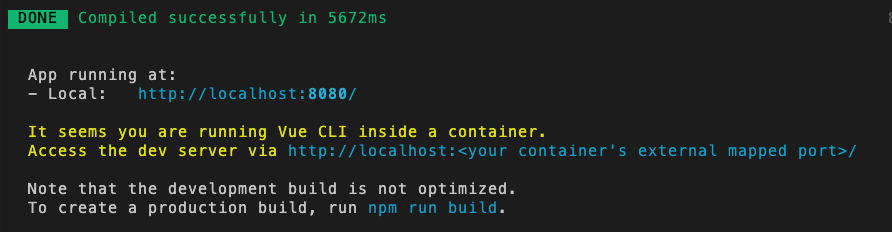
进入到 vue-next-test 目录下。

cd vue-next-test

现在我们使用以下命令，来运行一下项目吧！

npm run serve

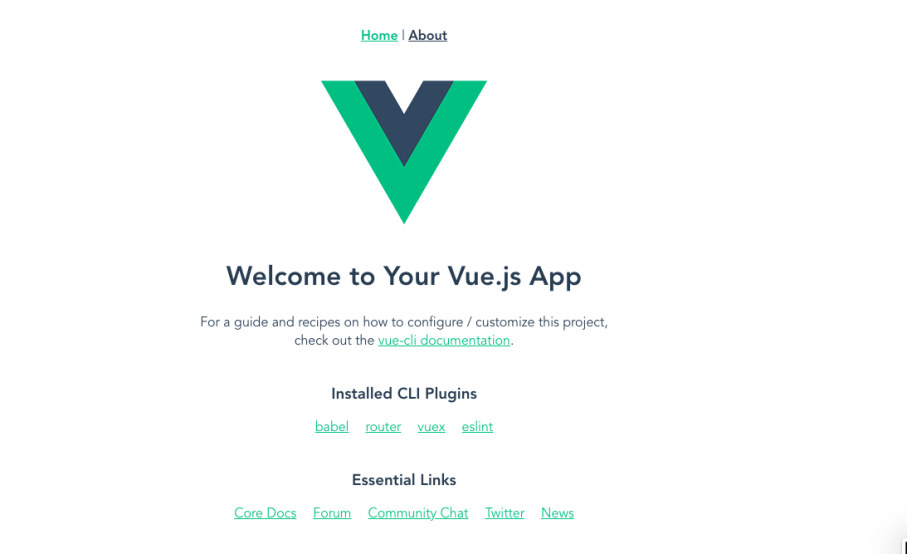
如果看到以下内容，说明项目运行成功。



运行成功后，我们点击右侧的 **Web 服务** 进行浏览。



打开浏览器我们会看到如下图所示的内容。



# 从“Hello World” 学习基础API

## Hello World

我们先举个简单的例子吧！使用 Vue 3 来输出一个 Hello World。在 /home/project 下新建一个名为 Hello.html 的文件，在该文件中写入以下代码：

<!--Hello.html-->

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<title>Hello</title>

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<meta http-equiv="X-UA-Compatible" content="ie=edge" />

<!-- 通过 cdn 方式引入 vue.js 的最新版本 -->

<script src="https://labfile.oss.aliyuncs.com/courses/2495/vue.global.js"></script>

</head>

<body>

<div id="app">

<p>{{msg}}</p>

<p>{{reversedMsg}}</p>

</div>

<script>

const { createApp, ref, computed } = Vue;

const App = {

setup() {

const msg = ref('Hello,World!');

const reversedMsg = computed(() =>

msg.value.split(',').reverse().join(',')

);

return {

msg,

reversedMsg,

};

},

};

createApp(App).mount('#app');

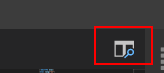
</script>

</body>

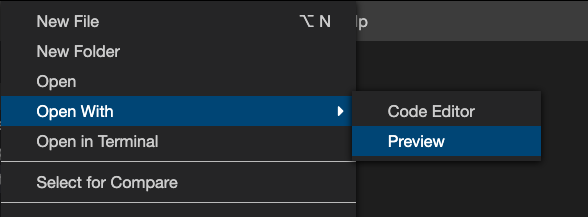
</html>

对于 .html 文件，在我们的环境中有如下两种查看方式：

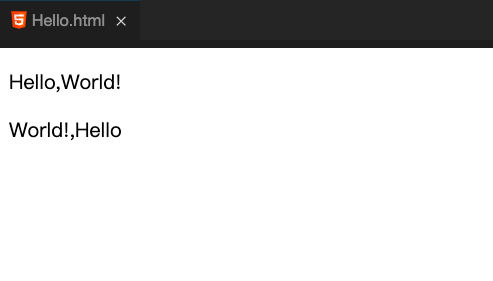
其一：点击代码框右上角的图标，即可查看。



其二：鼠标移到 Hello.html 文件上，右击选择 Open With -> Preview 也可以查看。



运行效果如下所示：



我们观察两个版本的实现方法，发现其中页面结构是完全相同的，但是 <script></script> 中的实现方式有一些不同：

1. Vue 3 使用 setup() 代替了 data() 以及 computed()。
2. Vue 3 使用了 ref() 方法。
3. Vue 3 使用 createApp() 代替了 new Vue({})。

接下来我们将分别为大家讲解这三点区别。