

实验 09：部署 Vue.js 开发环境

一、实验目的

- 1、掌握 Vue.js 开发环境的基本搭建方法，包括 Node.js 安装、Vue CLI 工具使用；
- 2、理解 Vue 项目结构，熟悉单文件组件(SFC)的基本概念和使用；
- 3、能够创建并运行 Vue 项目，掌握开发服务器的启动和项目构建流程；
- 4、了解 Vue Devtools 调试工具的安装与使用，提升开发效率。

二、实验学时

2 学时

三、实验类型

综合性

四、实验需求

1、硬件

每人配备计算机 1 台，建议优先使用个人计算机开展实验。

2、软件

安装 Visual Studio Code，以及 Edge 浏览器。

3、网络

本地主机能够访问互联网和实验中心网络。

4、工具

无。

五、实验任务

1、完成 Node.js 环境安装与配置

2、完成 Vue CLI 的安装与验证

3、创建第一个 Vue 项目并运行

4、了解项目结构并完成简单修改

六、实验内容及步骤

1、Node.js 环境安装与配置

步骤1：下载 Node.js 安装包，访问 Node.js 官网：<https://nodejs.org/zh-cn/download>，点击“Windows 安装程序(.msi)”，下载最新版 Node.js。

下载 Node.js®

获得适用于 Windows 且使用 Docker 和 npm 的 Node.js® v24.11.1 (LTS)

提示 Want new features sooner? Get the [latest Node.js version](#) instead and try the latest improvements!

```
1 # Docker 对每个操作系统都有特定的安装指导。
2 # 请参考 https://docker.com/get-started/ 给出的官方文档
3
4 # 拉取 Node.js Docker 镜像:
5 docker pull node:24-alpine
6
7 # 创建 Node.js 容器并启动一个 Shell 会话:
8 docker run -it --rm --entrypoint sh node:24-alpine
9
10 # 验证 Node.js 版本:
11 node -v # Should print "v24.11.1".
12
13 # 验证 npm 版本:
14 npm -v # Should print "11.6.2".
```

PowerShell 复制到剪贴板

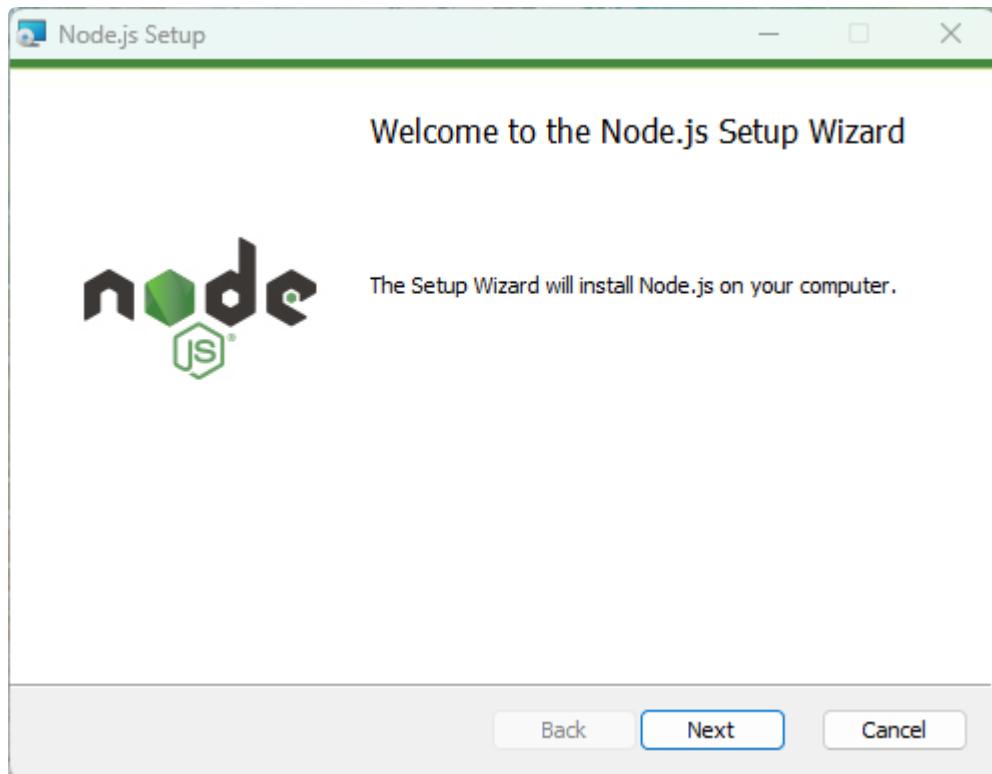
Docker 是一个容器化平台。如果遇到任何问题，请访问 [Docker的网站](#)

或者获得适用于 Windows x64 平台的 Node.js® 构建。

Windows 安装程序(.msi) 独立文件(zip)

阅读此版本的 [变更日志](#) 或 [博客文章](#)。
了解有关 [Node.js 发布](#) 的更多信息，包括发布时间表和长期支持版本的状态。
了解如何验证 [已签名的 SHASUMS](#)。
正在寻找 Node.js 的源代码？下载已签名的 [Node.js 源代码压缩包](#)。
检查 [nightly](#) 二进制文件或所有 [之前的版本](#) 或用于其他平台的 [非官方的](#) 二进制文件。

步骤2：双击下载的.msi文件，按向导完成安装。



步骤3：打开命令提示符，分别输入以下命令检查node和npm的版本。

Plain Text

```
1 node --version  
2 npm --version
```

2、Vue CLI 安装与验证

步骤1：全局安装Vue CLI

Plain Text

```
1 npm install -g @vue/cli
```

步骤2：验证Vue CLI安装

Plain Text

```
1 vue --version
```

步骤3：检查Vue CLI功能

Plain Text

```
1 vue --help
```

```
C:\Users\leile>vue --version
@vue/cli 5.0.9

C:\Users\leile>vue --help
Usage: vue <command> [options]

Options:
  -V, --version                                output the version number
  -h, --help                                     display help for command

Commands:
  create [options] <app-name>                  create a new project powered by vue-cli-service
  add [options] <plugin> [pluginOptions]        install a plugin and invoke its generator in an already created project
  invoke [options] <plugin> [pluginOptions]      invoke the generator of a plugin in an already created project
  inspect [options] [paths...]                   inspect the webpack config in a project with vue-cli-service
  serve                                         alias of "npm run serve" in the current project
  build                                         alias of "npm run build" in the current project
  ui [options]                                    start and open the vue-cli ui
  init [options] <template> <app-name>          generate a project from a remote template (legacy API, requires @vue/cli-init)
  config [options] [value]                      inspect and modify the config
  outdated [options]                           (experimental) check for outdated vue cli service / plugins
  upgrade [options] [plugin-name]              (experimental) upgrade vue cli service / plugins
  migrate [options] [plugin-name]             (experimental) run migrator for an already-installed cli plugin
  info                                           print debugging information about your environment
  help [command]                                display help for command

Run vue <command> --help for detailed usage of given command.

C:\Users\leile>
```

3、创建并运行第一个Vue项目

步骤1：在合适位置创建项目文件夹，如vue-projects，使用Visual Studio Code打开目录，并启动终端执行以下命令。

Plain Text

```
1 vue create my-first-vue-app
```

步骤2：按照提示完成项目创建

```
npm config get registry
Vue CLI v5.0.9
? Please pick a preset: (Use arrow keys)
> Default ([Vue 3] babel, eslint)
Default ([Vue 2] babel, eslint)
Manually select features
```

步骤3：使用以下命令运行项目

Plain Text

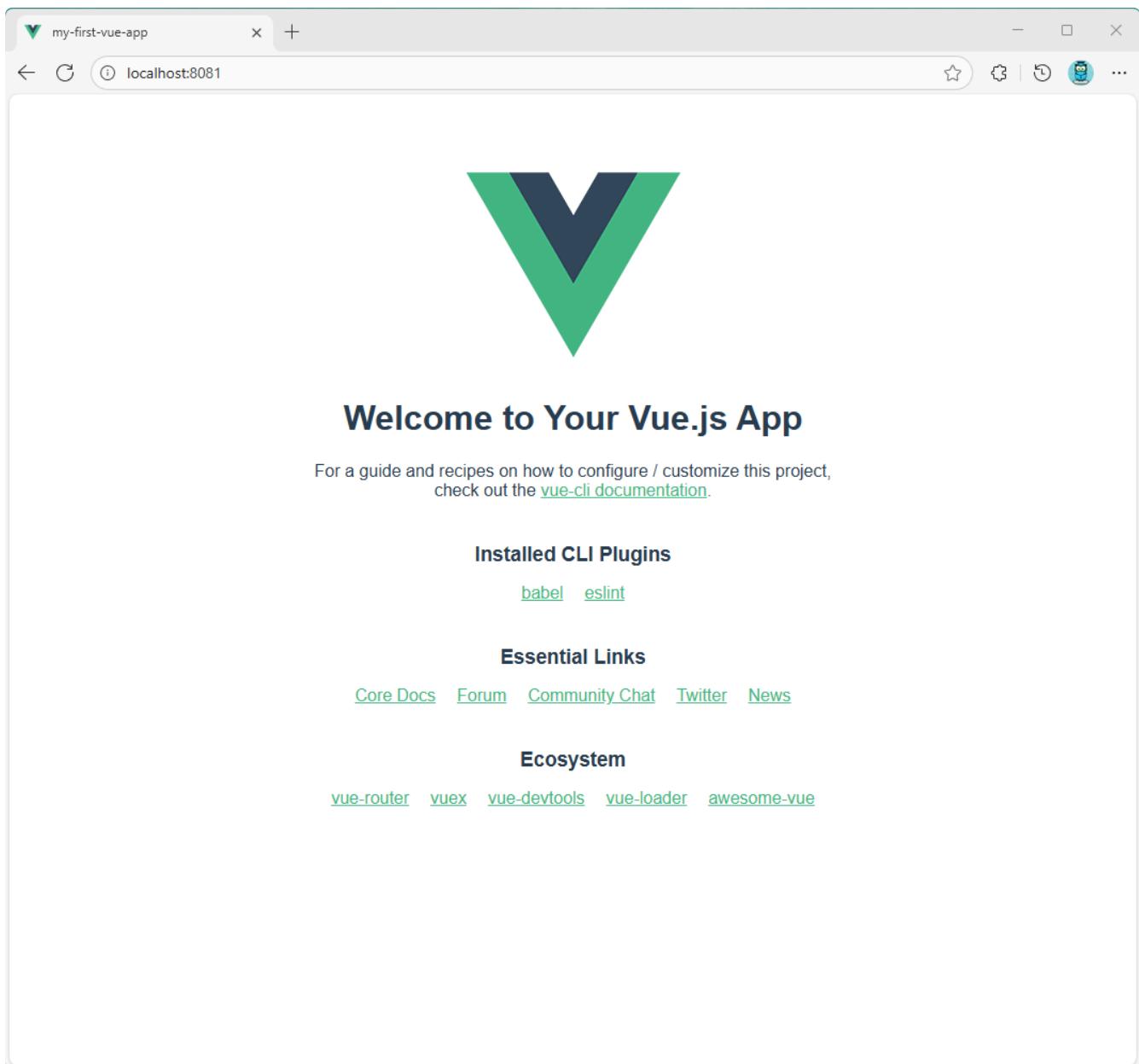
```
1 cd my-first-vue-app
2 npm run serve
```

```
选择 C:\WINDOWS\system32\cmd.exe

C:\Users\leile\Desktop\demo>cd my-first-vue-app
C:\Users\leile\Desktop\demo\my-first-vue-app>npm run serve
> my-first-vue-app@0.1.0 serve
> vue-cli-service serve
INFO Starting development server...
DONE Compiled successfully in 2854ms
22:10:05

App running at:
- Local: http://localhost:8081/
- Network: http://192.168.31.25:8081/
Note that the development build is not optimized.
To create a production build, run npm run build.
```

步骤4：访问创建好的页面



4、了解项目结构并完成修改

步骤1：分析项目结构，主要目录和文件如下。

Plain Text

```
1 my-first-vue-app/
2 └── public/          # 静态资源
3   ├── index.html      # 主HTML文件
4   └── favicon.ico    # 网站图标
5 └── src/             # 源代码目录
6   ├── assets/         # 资源文件(图片、样式等)
7   ├── components/     # Vue组件
8   │   └── App.vue      # 根组件
9   └── main.js          # 入口文件
10 └── package.json     # 项目配置和依赖
11 └── vue.config.js    # Vue特定配置(可选)
```

步骤 2：修改 App.vue 组件

打开 src/App.vue，找到 <template> 部分，修改欢迎信息。

XML/HTML

```
1 <template>
2   <div id="app">
3     
4     <h1>我的第一个Vue应用</h1>
5     <p>欢迎学习Vue.js开发! </p>
6     <!-- 添加自定义内容 -->
7     <div class="custom-section">
8       <h2>实验成果展示</h2>
9       <p>成功部署Vue开发环境! </p>
10    </div>
11  </div>
12 </template>
```

步骤 3：添加简单样式，在 <style> 部分添加 CSS。

XML/HTML

```
1 <style>
2 #app {
3   font-family: Avenir, Helvetica, Arial, sans-serif;
4   text-align: center;
5   color: #2c3e50;
6   margin-top: 60px;
7 }
8
9 .custom-section {
10   margin: 20px auto;
11   padding: 20px;
12   background-color: #f5f5f5;
13   border-radius: 8px;
14   max-width: 500px;
15 }
16
17 h1 {
18   color: #42b983;
19 }
20 </style>
```

步骤4：查看修改效果。

- 保存文件后，浏览器中的页面会自动刷新
- 观察新增的内容和样式变化

步骤5：尝试添加响应式数据

在<script>部分添加数据。

XML/HTML

```
1 <script>
2 export default {
3   name: 'App',
4   data() {
5     return {
6       message: '这是动态数据',
7       count: 0
8     }
9   },
10  methods: {
11    increment() {
12      this.count++
13    }
14  }
15 }
16 </script>
```

在模板中使用数据。

XML/HTML

```
1 <template>
2   <div id="app">
3     <!-- 已有内容 -->
4     <div class="interactive-section">
5       <p>{{ message }}</p>
6       <p>计数器: {{ count }}</p>
7       <button @click="increment">点击增加</button>
8     </div>
9   </div>
10 </template>
```

七、实验考核

本实验考核采用【实验随堂查】方式开展。

每个实验完成后，在实验课上通过现场演示的方式向实验指导教师进行汇报，并完成现场问答交流。

每个实验考核满分 100 分，其中实验成果汇报 60 分，现场提问交流 40 分。

实验考核流程：

- (1) 学生演示汇报实验内容的完成情况，实验指导老师现场打分。
- (2) 指导老师结合实验内容进行提问，每位学生提问 2-3 个问题，根据回答的情况现场打分。
- (3) 实验考核结束后，进行公布成绩。