搭建基础项目

更新时间：2019-07-24 14:09:33

> 为中华之崛起而读书。
>
> ——周恩来

这一章是实战章节，我们将会使用 Vue+TypeScript 来开发一个简单管理后台，这个实战项目的目的是演示如何在 Vue 项目中使用 TypeScript 进行开发，而不是为了实现一个管理后台。所以我们在开发的时候，以尽可能多的展示如何使用 TypeScript 结合 Vue 全家桶进行开发为前提，进行这个实战项目的开发，对于涉及到相同知识点的功能性代码，我们会尽可能不去重复讲解。

6.1.1 安装Vue CLI

本小节我们先来进行第一步，就是创建我们的初始项目，首先你需要在全局安装 Vue 命令行工具 Vue CLI，首先你要确定你安装了 NodeJS，如何安装我们已经在1.3小节讲过了，我们直接看如何安装 Vue CLI：

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

> Windows系统直接运行 `npm install -g @vue/cli` 命令即可。

因为是在全局安装，所以你可能需要以管理员身份去运行，Mac中就是前面加 `sudo` 然后输入密码。安装之后你可以通过检查他的版本号来确定是否成功安装：

```
vue -V 或者 vue --version
```

注意这里是使用 `vue` 命令，而不是@vue/cli，如果使用 `-V` 参数一定记着是大写的**V**，我这里安装的是**3.3.0**版本的，如果你发现你创建的项目和我创建的项目包含的文件或者配置信息有差异，有可能是版本差异造成的，你也可可以直接安装与我相同的版本。

6.1.2 使用Vue CLI创建项目

安装好Vue CLI后，在终端启动Vue的可视化项目管理页面，运行如下命令：

```
vue ui
```

几秒钟之后就会自动打开浏览器在浏览器中打开一个本地页面，然后打开创建项目界面：

选择要存放新项目的文件夹后，点击底部的“在此创建新项目”，然后进入配置流程，一共四个流程：“详情” - “预设” - “功能” - “配置”，我们来逐个配置。

(1) 详情

填写项目名，这里我们写的是"vue-ts-project"，然后选择包管理器，我这里选的是npm，如果你建了git仓库，你可以在git栏填写git的仓库地址。配置完成后点击“下一步”。

(2) 预设

每次创建一次项目，你的功能配置都可以保存为预设，方便下次直接使用，这里我们先选择手动，来手动配置项目。然后点击“下一步”。

(3) 功能

这里我们勾选这几项：

- Babel：使用Babel可以将新特性语法转为低版本浏览器支持的语法；
- TypeScript：这个肯定要勾选了，因为我们要使用TypeScript开发；
- Router：我们要开发单页面SPA应用，所以页面间的跳转需要使用前端路由；
- Vuex：多个页面直接或者组件之间的通信可以使用Vuex来做状态管理；
- CSS Pre-processors：CSS预处理器，如果你使用Less或者Sass等CSS预处理器编写CSS样式，则勾选这项；
- Linter/Formatter：代码规范或格式检查器，如果你需要使用TSLint对代码规范或格式进行检验，勾选这项；
- 使用配置文件：如果需要将一些插件的配置保存在各自的配置文件中，则需要勾选这项。

配置完成后点击“下一步”。

(4) 配置

这里有吉祥需要配置：

- Use class-style component syntax?

这项勾选上代表要在类风格的组件中使用@Component装饰器

- Use Babel alongside TypeScript for auto-detected polyfills?

勾选这一项后，会同时使用Babel和TypeScript对新标准语法进行转义。

- Pick a CSS pre-processor

这里我选择的是Less，你可以根据自己习惯勾选。

- Pick a linter / formatter config

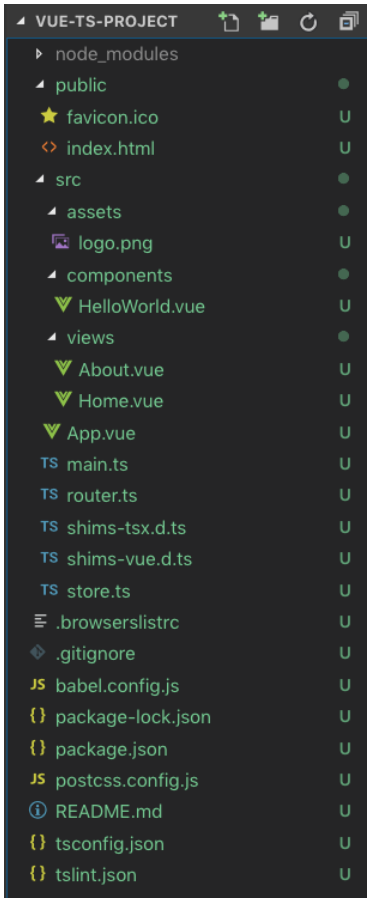
这里是选择代码检查工具，我们使用TypeScript进行开发，所以选择TSLint。

- Lint on save

勾选这一项后，会在文件保存后进行代码风格和格式的检验。

到这里配置就完成了，点击创建项目，Vue CLI会根据我们的配置生成配合文件，同时会帮我们安装好依赖。

6.1.3 项目结构介绍



下面我们来介绍下每个文件的作用：

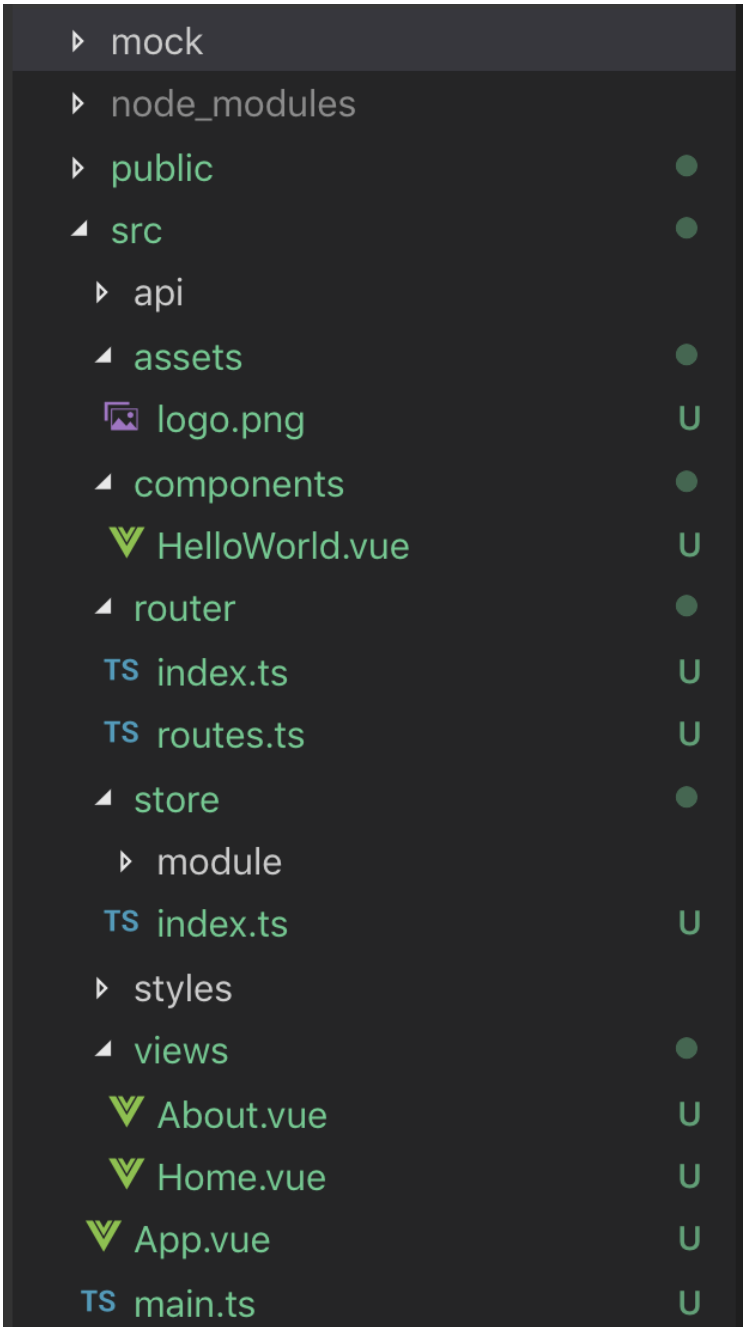
- public文件夹
  - favicon.ico： 显示在浏览器标签栏标题前面的小图标
  - index.html： 编译html文件的模板
- src
  - assets： 存放一些图片、字体等静态资源
  - components： 一些可以复用的组件存放在这里
  - views： 页面视图vue文件存放在这里
  - main.ts： 项目入口文件

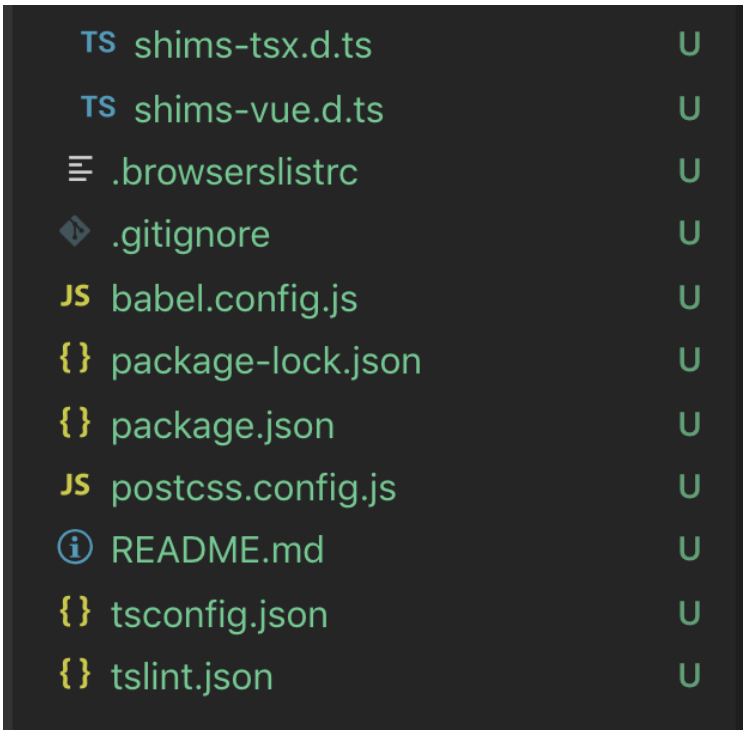
- router.ts： 前端路由配置文件
- shims-tsx.d.ts： 增加对JSX语法的类型支持的声明文件
- shims-vue.d.ts： 用于让编译器识别.vue后缀的文件
- store.ts： 状态管理vuex配置文件

- .brwserslistrc： 配置编译后的代码需要支持的浏览器列表
- .gitignore： 设置提交到git时需要忽略的文件
- babel.config.js： babel的配置文件
- package-lock.json： 锁定依赖版本号的文件
- package.json： npm项目最基本的配置文件
- postcss.config.js： postcss的配置文件
- README.md： 项目介绍文件
- tsconfig.json： 这个前面讲过了，TypeScript编译选项配置文件
- tslint.json： TSLint配置文件

6.1.4 调整项目结构

下面我们根据开发习惯调整一下目录结构，整理后的目录如下：





我们在根目录创建mock文件夹用于存放mock相关文件。在src文件夹创建api文件夹，用于存放封装的api请求方法；创建router文件夹，将原本src文件夹下的router.ts文件放在router文件夹下，并且改名为index.ts，然后把routes字段的数组抽出来放在routes.ts文件夹下默认导出：

```
// src/router/index.ts
import Vue from 'vue';
import Router from 'vue-router';
import routes from './routes';

Vue.use(Router);

export default new Router({
  routes,
});
// src/router/routes.ts
import Home from '../views/Home.vue'; // 注意这里的路径，有原来的./开头改为../，因为此时views文件夹在routes.ts文件所在文件夹的上一级

export default [
  {
    path: '/',
    name: 'home',
    component: Home,
  },
  {
    path: '/about',
    name: 'about',
    // route level code-splitting
    // this generates a separate chunk (about.[hash].js) for this route
    // which is lazy-loaded when the route is visited.
    component: () => import(/* webpackChunkName: "about" */ '../views/About.vue'),
  },
]
```

在src文件夹下创建store文件夹，把原本在src文件夹下的store.ts文件放在store文件夹下，并改名为index.ts，然后创建module文件夹，用于存放拆分的vuex模块；在src文件夹创建styles文件夹，用于存放公共的样式文件；修改main.ts文件里对router和store的引用。

我们还需要在根目录创建一个*vue.config.js*文件，用来配置vue项目相关配置，文件内的内容如下：

```
const path = require('path')

const resolve = dir => {
  return path.join(__dirname, dir)
}

const BASE_URL = process.env.NODE_ENV === 'production'
  ? './'
  : '/'

module.exports = {
  publicPath: BASE_URL, // 公共文件路径
  lintOnSave: true, // 在保存文件的时候对代码进行格式校验
  chainWebpack: config => {
    config.resolve.alias
      .set('@', resolve('src')) // 配置便捷路径，凡是src这一级路径都可以用@代替
  }
}
```

这样我们的项目就算基本创建完成，后面我们会通过具体的功能开发，讲解vue全家桶的TypeScript开发用法。现在你可以通过如下命令启动项目：`npm run serve`启动后打开终端显示的url在浏览器打开项目页面。下个小节我们来封装axios，用它来进行接口调用。

书写声明文件之砍柴：为不同类型库书写声明文件

封装接口请求

精选留言 0

欢迎在这里发表留言，作者筛选后可公开显示

目前暂无任何讨论