

# Vue 项目搭建

旺财记账

# 版权声明

本内容版权属杭州饥人谷教育科技有限公司（简称饥人谷）所有。

任何媒体、网站或个人未经本网协议授权不得转载、链接、转贴，或以其他方式复制、发布和发表。

已获得饥人谷授权的媒体、网站或个人在使用时须注明「资料来源：饥人谷」。

对于违反者，饥人谷将依法追究其责任。

# 联系方式

如果你想要购买本课程

请微信联系 [xiedaimala02](#) 或 [xiedaimala03](#)

如果你发现有人盗用本课程

请微信联系 [xiedaimala02](#) 或 [xiedaimala03](#)

# 目录

- 用 @vue/cli 创建项目
- 用 VueRouter 添加底部导航
- 引入 SVG Icon
- 部署到GitHub和码云

# 课前准备

- 请检查以下前置项

- ✓ 全局安装 @vue/cli@4.1.2 以上版本
- ✓ 安装 Node.js v10 以上版本
- ✓ VSCode 最新版或者 WebStorm 最新版均可
- ✓ 科学上网，能访问 Google

如果你是前端体系班学生，那么这些你都已经准备好了

# 创建项目

- 步骤

- ✓ vue create morney （项目名可以自定）
- ✓ cd morney
- ✓ yarn serve

# vue create 配置 1

```
Vue CLI v4.1.2
```

```
? Please pick a preset:  
  default (babel, eslint)  
> Manually select features
```

```
? Check the features needed for your project:  
(* ) Babel  
(* ) TypeScript  
(* ) Progressive Web App (PWA) Support  
(* ) Router  
(* ) Vuex  
(* ) CSS Pre-processors  
(* ) Linter / Formatter  
(* ) Unit Testing  
> ( ) E2E Testing
```

```
? Use class-style component syntax? (Y/n) y
```

# vue create 配置 2

```
? Use Babel alongside TypeScript (required)? (Y/n) y
```

```
? Use history mode for router?  
n
```

```
? Pick a CSS pre-processor (PostCSS row keys)  
> Sass/SCSS (with dart-sass)  
  Sass/SCSS (with node-sass)  
  Less  
  Stylus
```

```
? Pick a linter / formatter config: Basic  
? Pick additional lint features:  
> ( ) Lint on save  
  (*) Lint and fix on commit
```



# vue create 配置 3

```
? Pick a unit testing solution:  
  Mocha + Chai  
> Jest
```

```
? Where do you prefer placing config for Babel, ESLint, etc.? In dedicated config files  
? Save this as a preset for future projects? (y/N) n
```

# vue create 完整配置

```
Vue CLI v4.1.2
```

```
? Please pick a preset: Manually select features
```

```
? Check the features needed for your project: Babel, TS, PWA, Router, Vuex, CSS Pre-processors, Linter, Unit
```

```
? Use class-style component syntax? Yes
```

```
? Use Babel alongside TypeScript (required for modern mode, auto-detected polyfills, transpiling JSX)? Yes
```

```
? Use history mode for router? (Requires proper server setup for index fallback in production) No
```

```
? Pick a CSS pre-processor (PostCSS, Autoprefixer and CSS Modules are supported by default): Sass/SCSS (with dart-sass)
```

```
? Pick a linter / formatter config: Basic
```

```
? Pick additional lint features: Lint and fix on commit
```

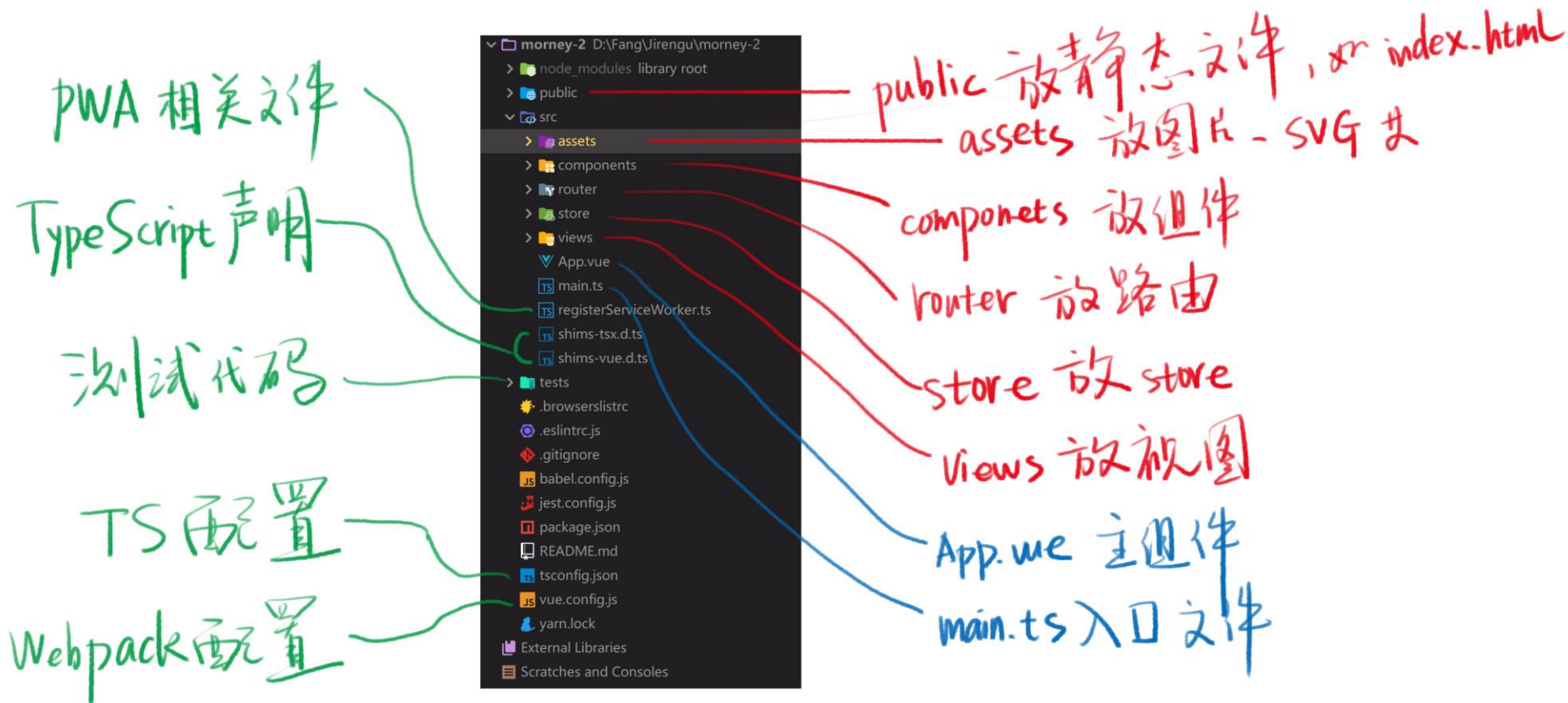
```
? Pick a unit testing solution: Jest
```

```
? Where do you prefer placing config for Babel, ESLint, etc.? In dedicated config files
```

```
? Save this as a preset for future projects? (y/N) n
```

注意：每个项目的配置是不同的，不要笼统地套用

# 目录说明



# 一些优化

- 编辑器配置

- ✓ 添加 vue snippets (分别演示 WebStorm 和 VSCode)
- ✓ VSCode 安装 Vetur 和 Vue VSCode Snippets

- import alias

- ✓ TS/JS 可以通过 @/目录名 引入文件
- ✓ CSS/SASS 可以通过 ~@/目录名 引入文件
- ✓ 这样就不用自己计算相对路径了
- ✓ WebStorm 需要在设置里搜索 webpack 然后填上路径 D:\Fang\Jirengu\morney\node\_modules\@vue\cli-service\webpack.config.js, 而 VSCode 不需要配置

# 底部导航

使用 Vue Router

# 思路

- 一、确定页面 url

- ✓ #/money 记账
- ✓ #/labels 标签
- ✓ #/statistics 统计
- ✓ 默认进入 #/money
- ✓ 添加一个 404 页面

- 添加代码

- ✓ router/index.ts 添加 router，配置 4 个路径对应组件
- ✓ 初始化组件：Money.vue / Labels.vue / Statistics.vue
- ✓ 将 router 传给 new Vue()
- ✓ 在 App 组件里用 <router-view/> 给出 router 渲染区域

# 思路

- 二、展示 <Nav />

- ✓ 两种思路：在 <App/> 全局展示或每个组件自己引入
- ✓ 分析两种思路的利弊，最终选择组件自己引入
- ✓ 优化：使用 Vue.component 全局组件
- ✓ 确定思路后，添加 CSS
- ✓ CSS 也有两种思路：fixed 和 flex，各有利弊
- ✓ 做项目，需要权衡利弊，错了就改，没错就继续用

- 优化

- ✓ 事不过三，将布局抽离成组件

# 更多内容请看视频和代码

PPT 到此结束