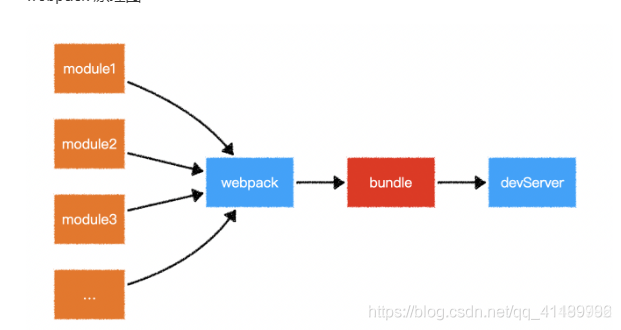
**Vite实用教程**

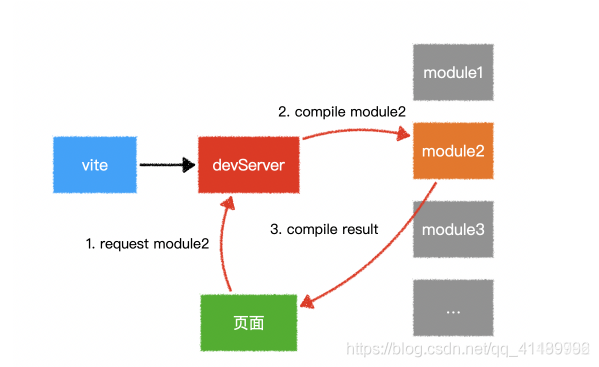
**一、vite与webpack区别（重点）**

1、vite轻快的冷服务启动，不需要打包，webpack启动时，将模块全部打包

webpack原理图：



vite原理图：



2、vite采用ES Module，webpack采用CommonJs

浏览器支持ES Module，会自动向依赖的module发出请求，开发环境vite将模块作为浏览器引用的文件；

3、vite按需加载，webpack全部加载

热更新：

1> vite只重新编译该模块（热更新原理：通过websocket，服务端将变更消息同步到客户端，客户端发送模块请求，重新编译模块并呈现）

2> webpack将模块的相关联模块全部编译一次；（HotModuleReplacementPlugin）

4、vite的优势在开发环境，生产环境采用传统rollup进行打包；

**一、Vite简介和环境搭建**

**1、简介**

它和vue-cli功能相同，都是项目初始化构建工具，vite属于vue-cli的第二代升级版，同时包含了项目的编译功能，vite在生产环境打包是通过Rollup完成的；

Rollup是JavaScript的模块bundler（打包器），可以将一小段代码打包成更大、更复杂的内容，例如库或应用程序；

即Vite（优势在开发环境）提供的是开发环境的编译，打包工作依靠的是Rollup；

1. **Vite特性**
2. Vite主打特点就是轻快冷服务启动。冷服务的意思是，在开发预览中，它是不进行打包的。
3. 热更新更快，模块内容变化时，只重新编译该模块，webpack编译所有依赖模块
4. 更好的支持typescript；
5. **Vite构建项目**

注意：目前vite只支持vue 3.x版，不支持vue 2.x版

1. yarn初始化项目

yarn create vite <project-name>

cd <project-name>

yarn dev

1. npm初始化项目

npm init vite@latest <project-name>

cd <project-name>

npm run dev

1. pnpm初始化项目

pnpm create vite <project-name>

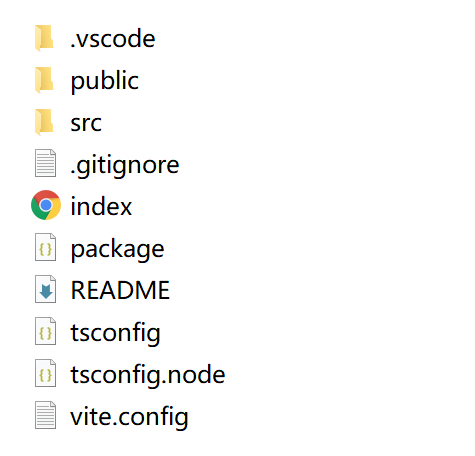
cd <project-name>

npm run dev

根据框架构建模板，命令如下：



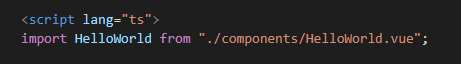
**二、项目结构**



**三、Vite对typescript、css、json的支持**

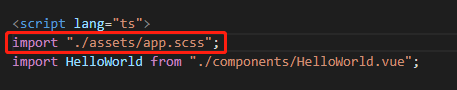
**1、对typescript的支持**

直接在组件script标签上添加lang=”ts”

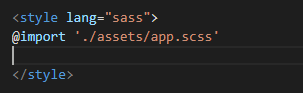


**2、对css的支持**

可以直接在.vue文件中，通过import引入css文件

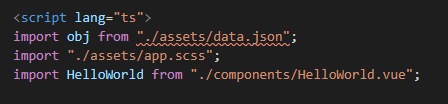


之前通过@import方式引入



**3、对json的支持**

可以直接在.vue文件中，通过import引入.json文件

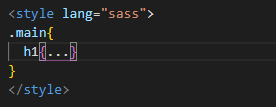


**五、Vite对sass和jsx的支持**

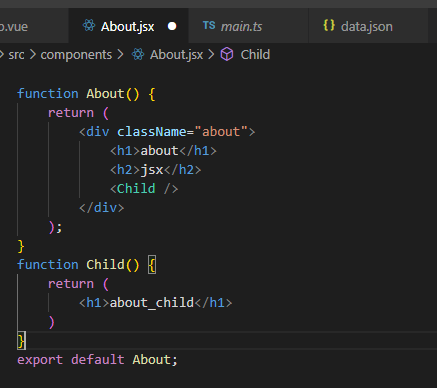
**1、安装sass支持**

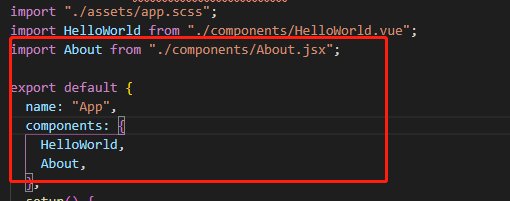
yarn add sass -D

直接组件style中定义lang=”sass”



1. **对jsx的支持**
2. 无状态jsx





1. 有状态jsx

  
**（不带事件绑定数据）**



**（带事件绑定修改数据）**



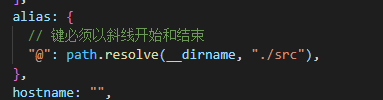
**六、Vite配置文件及别名设置**

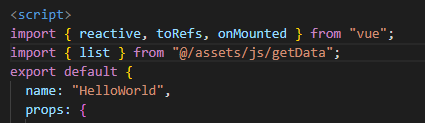
**1、vite配置文件**

在项目根目录新建一个vite.config.js或vite.config.ts文件，进行内容配置即可；

**2、别名配置（已正常） 最新版vite**

1、const { resolve } = require('path');





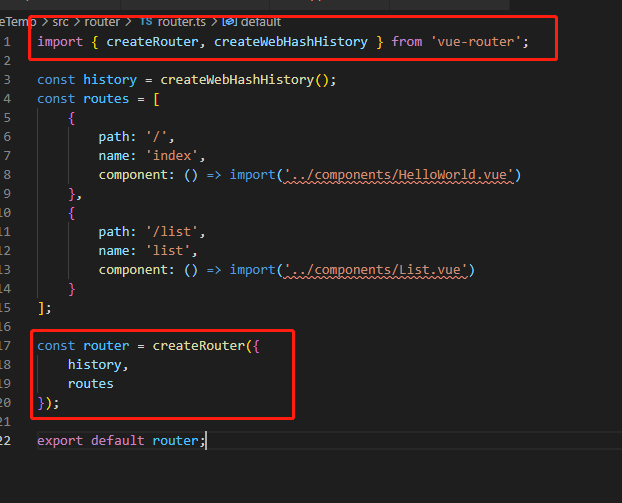
**七、Vite项目引入Vue-Router**

**1、安装router**

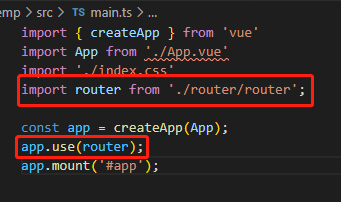
yarn add vue-router@next(@next安装最新版router，router4.0及以上版本才支持vue3)

**2、创建router.ts文件和main.ts引入路由**

1> 创建router.ts路由信息



2> main.ts引入router.ts路由

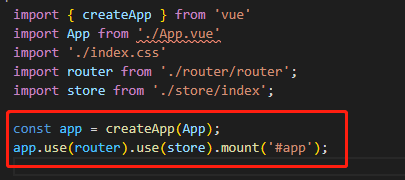


**八、Vite项目引入Vuex**

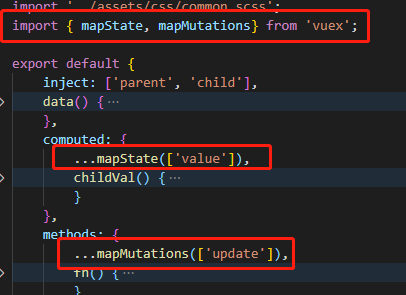
**1、安装Vuex  
 yarn add vuex@next -S**

**2、创建store/index.ts，配置vuex** 

**3、main.ts引入store**



**4、vuex状态数据操作**



**九、provide和inject**

**1、类型  
 provide**：Object | () => Object

**inject**：Array<string> | Object

例子：

provide: { a: 1, b: 2 } 或 provide() { return { a: 1, b: 2} }  
 inject: [‘a’, ‘b’] 或 inject: {a: {default: 11}, m: {from: ‘b’, default: 22}}

1. **动态数据绑定  
    provide设置**



**inject设置**

