

Vite

替代 Webpack 的下一代开发打包工具

康旭峰 2020-10-14



Native-ESM powered web dev build tool. It's fast.

– Evan You

Webpack 为什么慢？

- 模块分析装 生成 bundle
- 静态的打包，无论当前是否执行都打包
- 项目越来越大 生成 bundle 越来越慢



Vite 运行原理

- 基于 Native ES Modules
- 在 script 标签里设置 type="module"
- 在浏览器端使用 export、import
- 不需生成 bundle，天然按需加载



How fast?

Show Demo

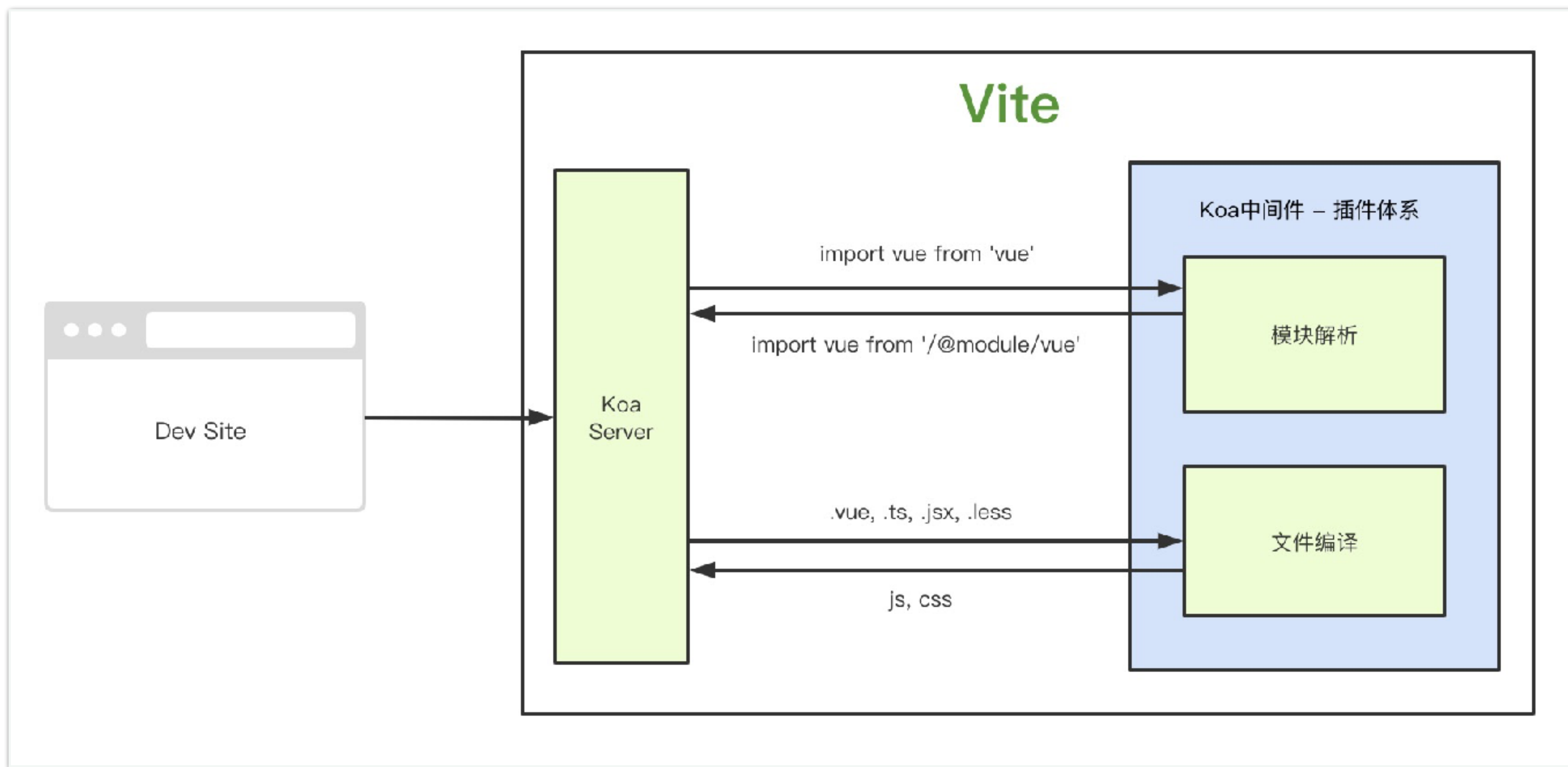
Native ES Modules

IE	Edge [*]	Firefox	Chrome	Safari	Opera	iOS Safari [*]	Opera Mini [*]	Android Browser [*]
	12-14							
	^{1 6} 15	2-53	4-59	3.1-10	10-46	3.2-10.2		
	⁶ 16-18	² 54-59	¹ 60	^{4 5} 10.1	¹ 47	^{4 5} 10.3		
6-10	79-85	60-80	61-85	11-13.1	48-70	11-13.7		2.1-4.4.4
11	86	81	86	14	71	14	all	81
		82-83	87-89	TP				

主流浏览器均已支持，开发模式可用

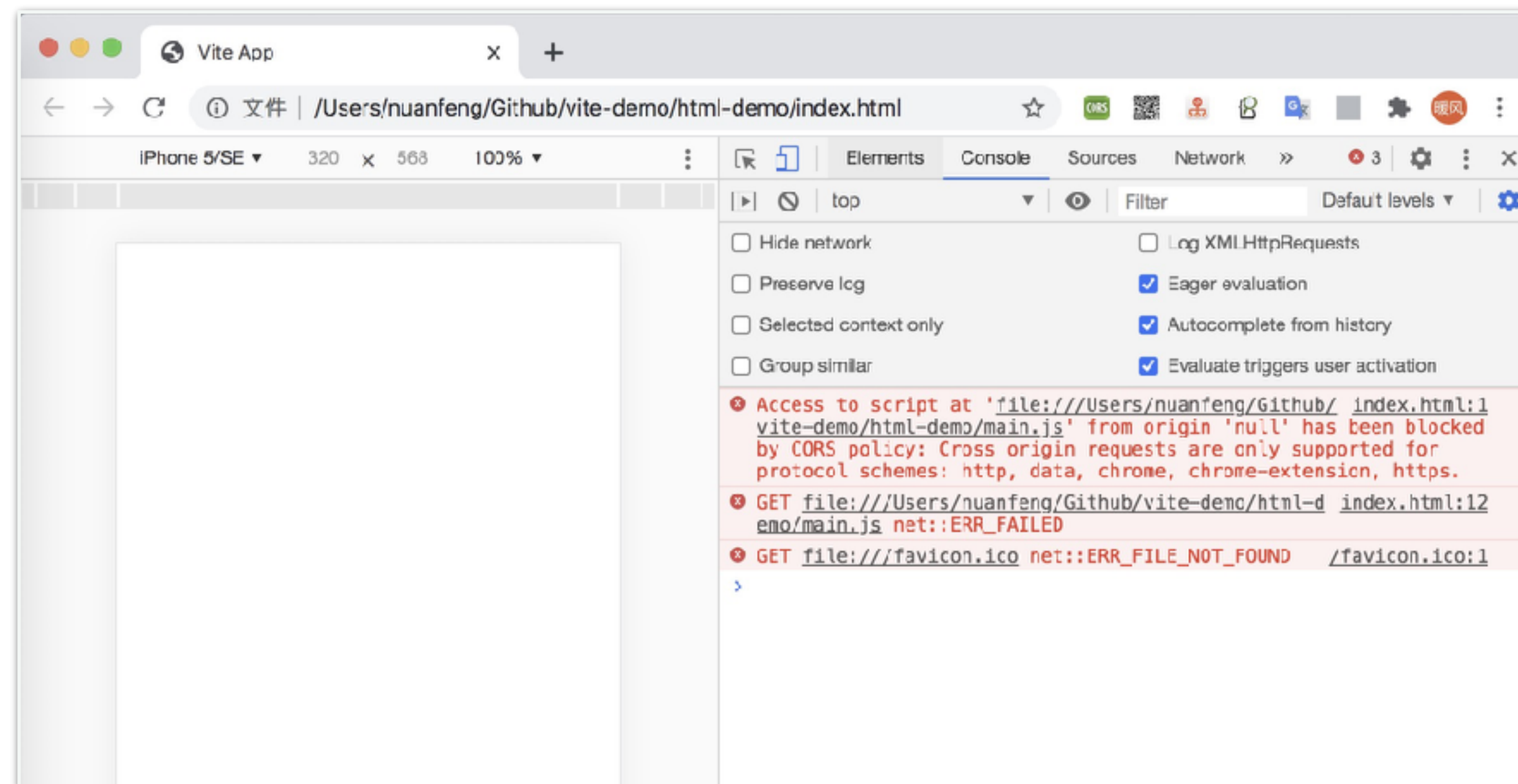
Vite 做了什么

1. 提供 web server
2. 模块解析
3. 文件编译



Vite 系统架构

1. 提供 web server



浏览器 ESM 文件不支持本地访问

1. 提供 web server

```
const { createServer } = require('vite')

const myPlugin = ({
  root, // project root directory, absolute path
  app, // Koa app instance
  server, // raw http server instance
  watcher // chokidar file watcher instance
}) => {
  app.use(async (ctx, next) => {
    if (ctx.path.endsWith('.scss')) {
      console.log('pre processing: ', ctx.url)
      ctx.type = 'css'
      ctx.body = 'body { border: 1px solid red }'
    }

    await next()

    if (ctx.response.is('js')) {
      console.log('post processing: ', ctx.url)
      console.log(ctx.body)
      // can be string or Readable stream
    }
  })
}

createServer({
  configureServer: [myPlugin]
}).listen(3000)
```

提供开放 Api, 使用灵活可自定义

2. 模块解析

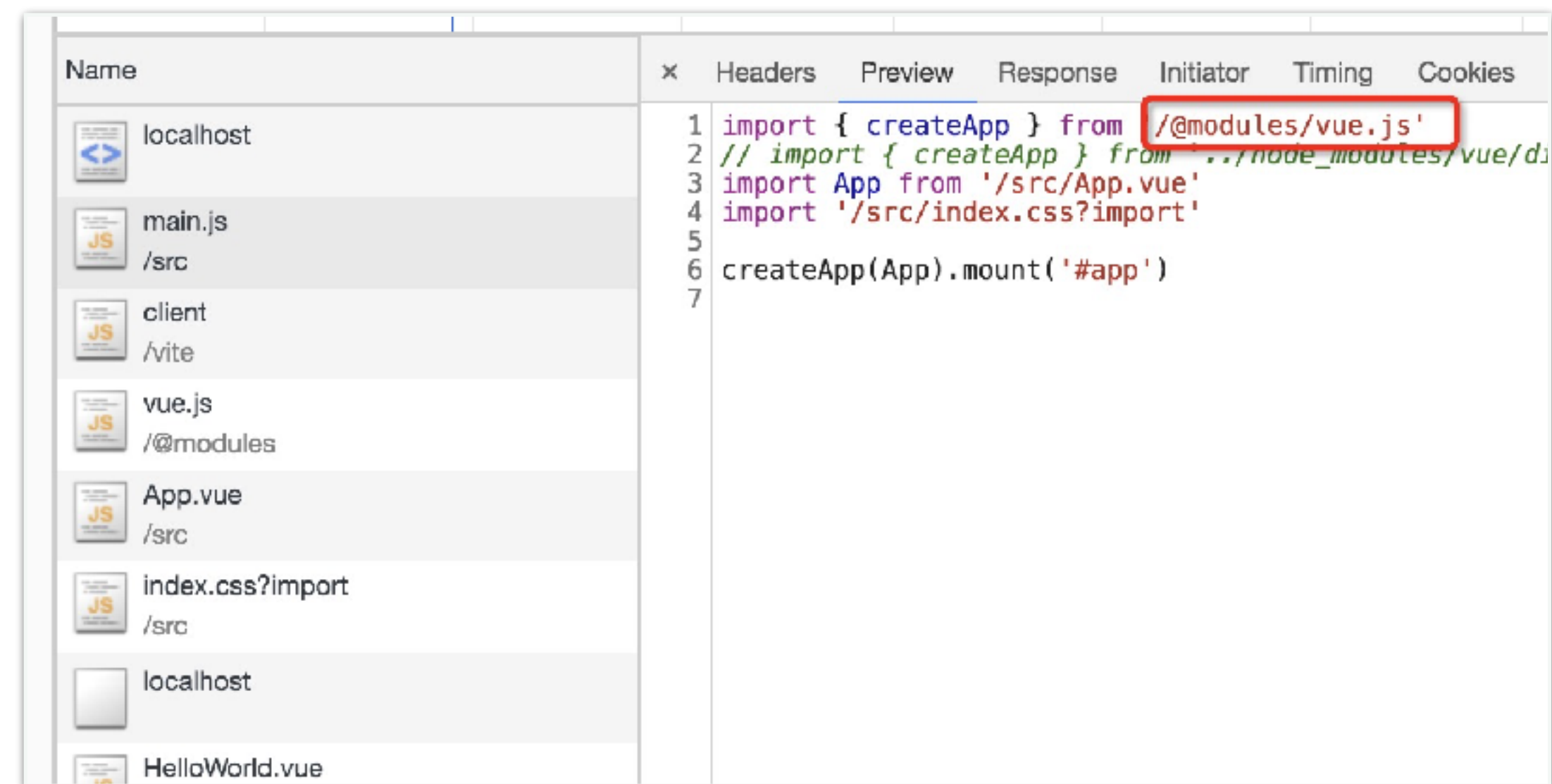
```
import { createApp } from 'vue'  
import App from './App.vue'  
import './index.css'  
  
createApp(App).mount('#app')
```

✖ Uncaught TypeError: Failed to resolve module specifier "vue". Relative references must start with either "/", "./", or "../". 127.0.0.1/:1

>

浏览器无法识别 vue 是在 node_modules 中的

2. 模块解析

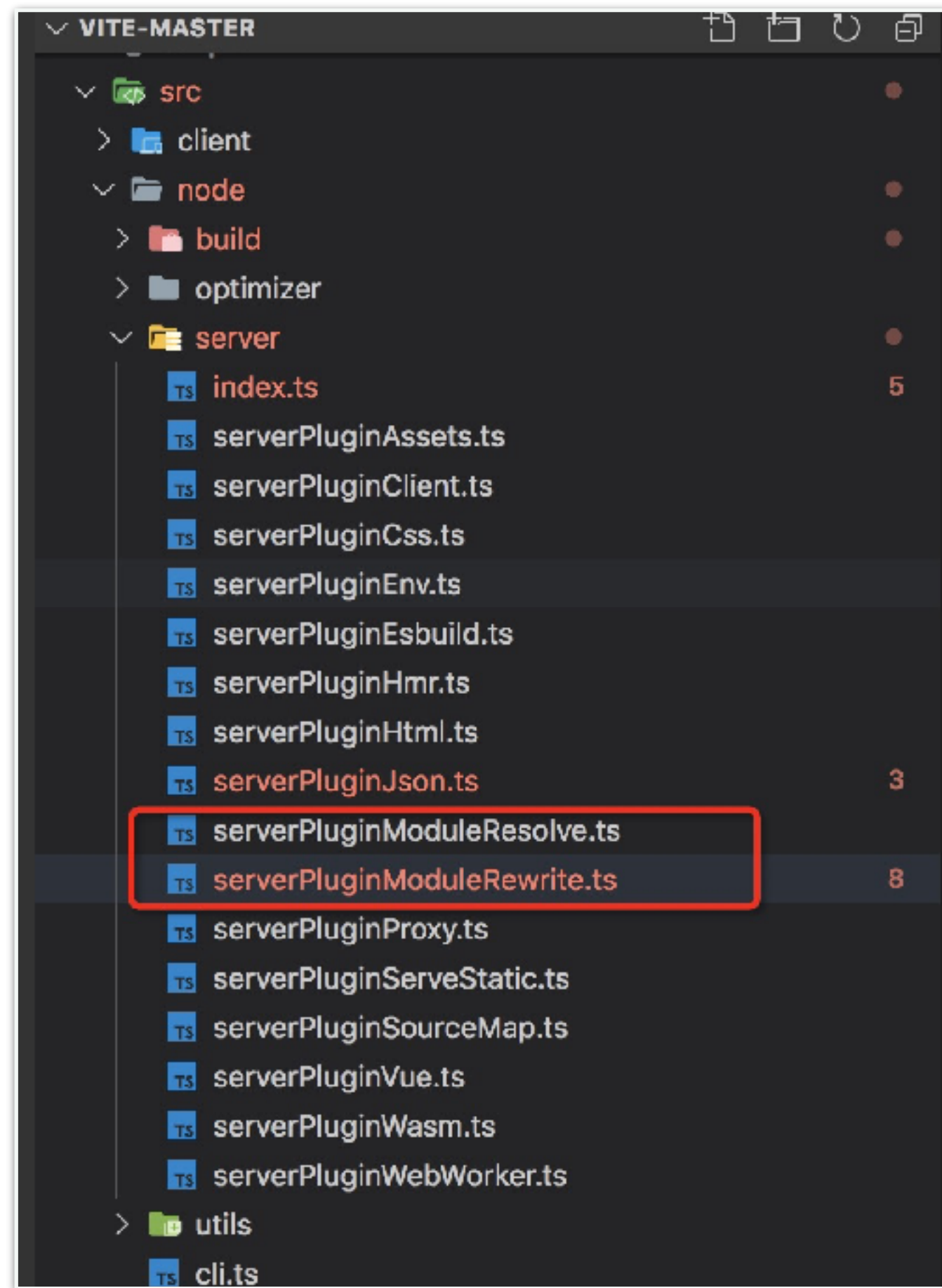


Vite 通过模块解析，将 `import 'vue'` 转换成了 `import '@modules/vue.js'`

2. 模块解析

1. 在 koa 中间件里获取请求 body
2. 通过 es-module-lexer 解析资源 AST 拿到 import 的内容
3. 判断 import 的资源是否是绝对路径，绝对视为 npm 模块
4. 返回处理后的资源路径: "vue" => "/@modules/vue"

2. 模块解析



moduleRewrite 负责重写路径

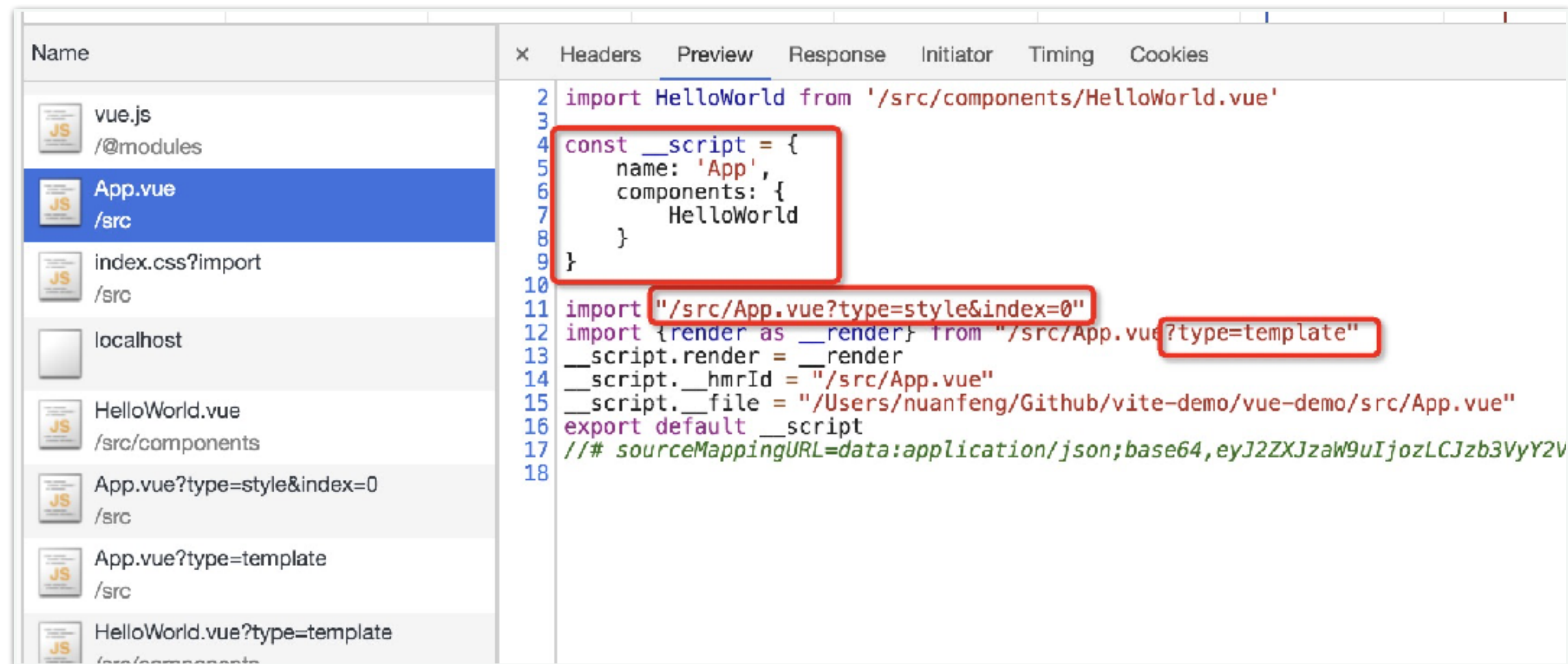
moduleResolve 负责匹配返回文件内容

在下一个 koa middleware 中，用正则匹配到路径上带有 @modules 的资源，再通过 `require('xxx')` 拿到包的内容导出返回给浏览器。

3.文件编译

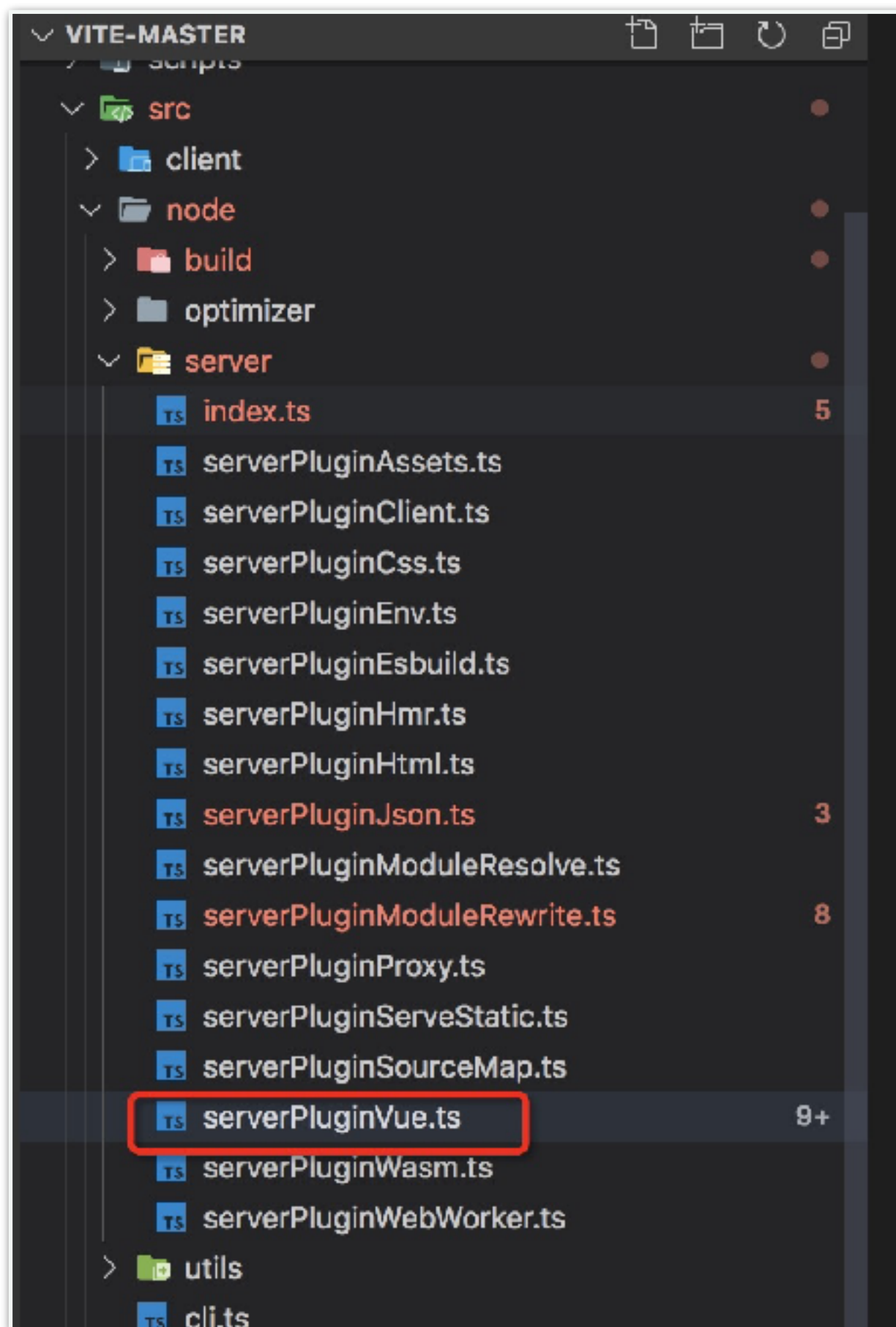
- .vue、.less、.jsx 等文件如何解析？
- webpack 中有对应的各种 loader
- Vite 中通过 Plugin 对各类文件类型做预处理

3.文件编译



.vue 文件返回前，被编译成 script 部分，和 style、template 两个独立的请求

3.文件编译



```
function vuePlugin({app}){
  app.use(async (ctx, next) => {
    if (!ctx.path.endsWith('.vue') && !ctx.vue) {
      return next()
    }

    const query = ctx.query
    // 获取文件名称
    let filename = resolver.requestToFile(publicPath)

    // 解析器解析 SFC
    const descriptor = await parseSFC(root, filename, ctx.body)
    if (!descriptor) {
      ctx.status = 404
      return
    }
    // 第一次请求 .vue
    if (!query.type) {
      if (descriptor.script && descriptor.script.src) {
        filename = await resolveSrcImport(descriptor.script, ctx, resolver)
      }
      ctx.type = 'js'
      // body 返回解析后的代码
      ctx.body = await compileSFCMain(descriptor, filename, publicPath)
    }

    // ...
  })
}
```

Vite 能在 React 项目使用吗?

<https://github.com/vitejs/vite-plugin-react>

Vite

- Vite 快是因为使用浏览器 ES Module 省略了 bundle 打包
- 通过拦截请求，做到了实时按需编译
- vue3 官方推荐，未来可期

<https://github.com/vitejs/awesome-vite>

Thanks