

#按需引用

我们可以只打包使用的组件或api，需要使用到 [unplugin-auto-import \(opens new window\)](#)和 [unplugin-vue-components \(opens new window\)](#)插件。

下面是vite.config.ts配置文件中 plugins内容，这也是向军大叔写的[前端脚手架 \(opens new window\)](#)的代码。

实现了以下功能

- element-plus的api与组件按需要加载
- 定义iconpark图标按需加载
- composables目录文件按需加载
- 项目 src/components 目录组件按自动加载

```
1 import { Plugin } from 'vite'
2 import AutoImport from 'unplugin-auto-import/vite'
3 import Components from 'unplugin-vue-components/vite'
4 import { ElementPlusResolver, VueUseComponentsResolver } from 'unplugin-vue-components/resolvers'
5
6 export default function autoImport(plugins: Plugin[]) {
7   plugins.push(
8     AutoImport({
9     resolvers: [ElementPlusResolver()],
10    //定义element-plus api按需加载
11    imports: ['vue', 'vue-router'],
12    //composables目录文件按需加载
13    dirs: ['src/composables'],
14    dts: 'types/auto-imports.d.ts',
15  })),
16  Components({
17    resolvers: [
18      //element-plus组件按需导入
19      ElementPlusResolver(),
20      VueUseComponentsResolver(),
21      //针对iconpark图标按需导入
22      (componentName) => {
23        if (componentName.startsWith('Icon')) {
24          return { name: componentName.slice(4), from: '@icon-park/vue-next' }
25        }
26      }
27    ]
28  })
29 }
```

```

26     },
27   ],
28   extensions: ['vue'],
29   dirs: ['src/components'],
30   //组件名称包含目录，防止同名组件冲突
31   directoryAsNamespace: true,
32   //指定类型声明文件，为true时在项目根目录创建
33   dts: 'types/components.d.ts',
34 },
35 )
36 }
37

```

如果element-plus使用按需要加载，我们可以将css文件定义在vue模板index.html中

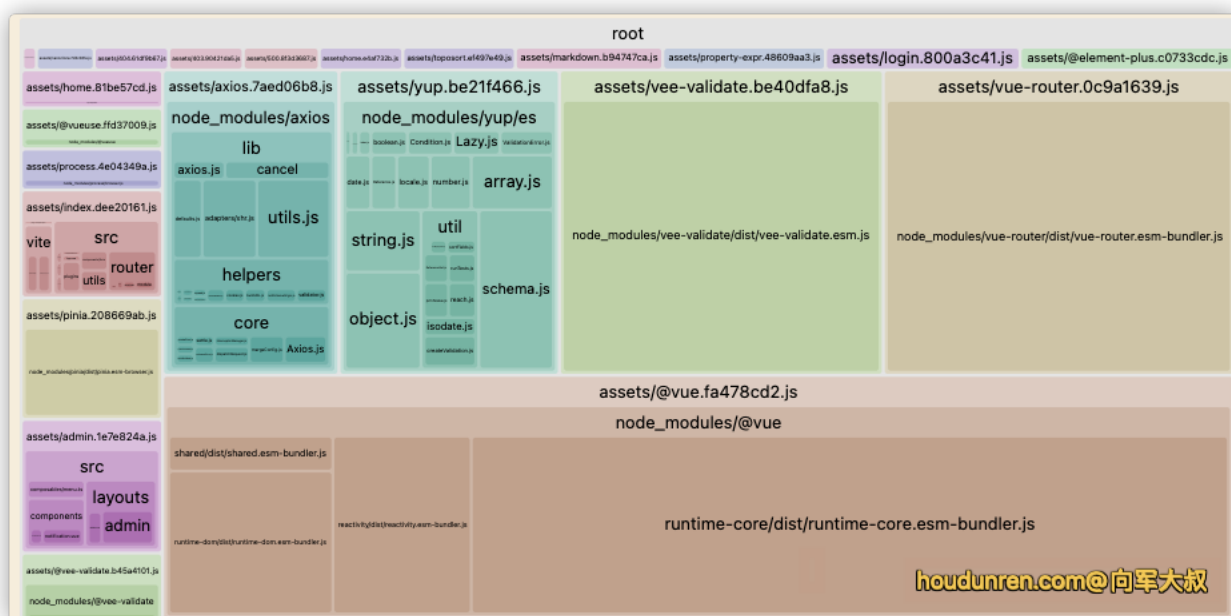
```

1  <link rel="stylesheet" href="//unpkg.com/element-plus/dist/index.css" />
2

```

#可视化

使用 [rollup-plugin-visualizer \(opens new window\)](#)可以对打包结果进行可视化的展示。



#安装配置

```
1 yarn add --dev rollup-plugin-visualizer
2
```

修改 `vite.config.ts` 配置文件

```
1 import { visualizer } from 'rollup-plugin-visualizer';
2
3 //...
4 plugins: [
5   // other plugins
6   // ....
7   //
8   visualizer()
9 ],
10 //...
11
```

#使用

下面执行编译命令

```
1 pnpm run build
2
```

这时会在项目根目录生成 `stats.html` 文件，使用浏览器打开这个文件即可查看到可视化的编译结果。

#分块打包

使用分块打包可以将内容打包到不同的文件中，减少单个文件大小，提高加载速度。下面将 `node_modules` 的扩展单独进行打包。

```
1 export default defineConfig({
2   plugins: [vue()],
3   build: {
4     rollupOptions: {
5       emptyOutDir: true,
6       output: {
```

```
7     manualChunks(id) {
8         if (id.includes('node_modules')) {
9             return id().split('node_modules/')[1].split('/')[0].()
10        }
11    },
12 },
13 },
14 },
15 })
16
```

#CDN加载

可以使用cdn加载的包，建议使用cdn完成，这样会减少打包后的文件大小。

下面是使用cdn加载 echarts、font-awesome、animate.css包。

```
1 <script src="
  https://cdn.bootcdn.net/ajax/libs/echarts/5.2.2/echarts.min.js
  <script src="
2 <link href="
  https://cdn.bootcdn.net/ajax/libs/font-awesome/6.0.0-
  beta3/css/all.min.css
  <link href="
3 <link href="
  https://cdn.bootcdn.net/ajax/libs/animate.css/4.1.1/animate.min.css
  <link href="
4
```

如果使用ts编译代码，不要忘记安装 echarts的类型支持

```
1 yarn add @types/echarts
2
```