

可编程组件

[可编程组件使用@antv/l7 指南](#)

[创建可编程组件工程，（可直接使用此项目）](#)

[可编程组件开发](#)

可编程组件使用@antv/l7 指南

ps:目前可编程组件不支持带斜杠的包，需要额外修改，方法见下文

注：本文假定已配置好开发环境，并熟悉可编程组件开发流程

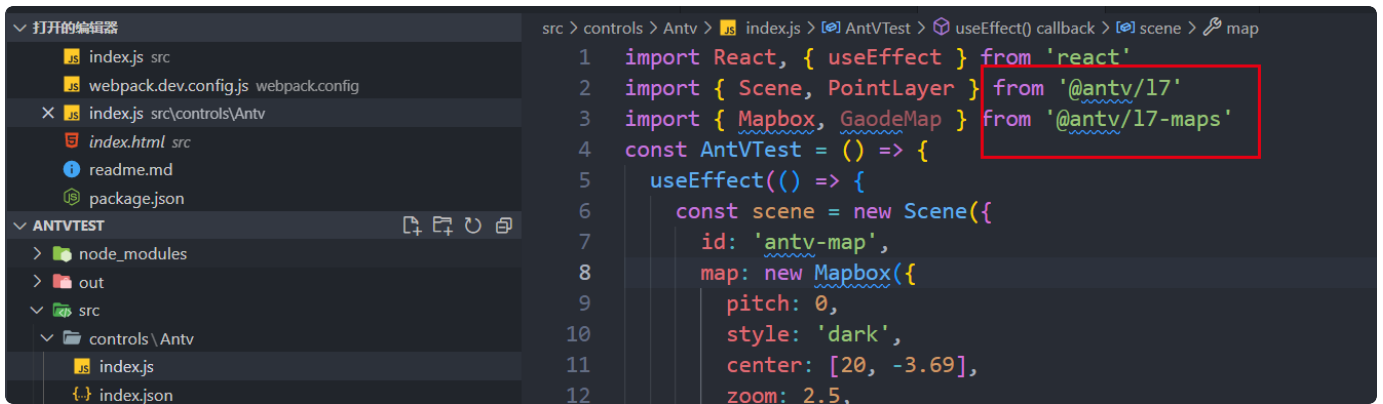
创建可编程组件工程，（可直接使用此项目）

修改.suposrc.json 配置文件

```
| server | supos 地址 |  
| username | supos 管理员账号 |  
| password | supos 管理员密码 |  
| app | 业务设计器中 appId |
```

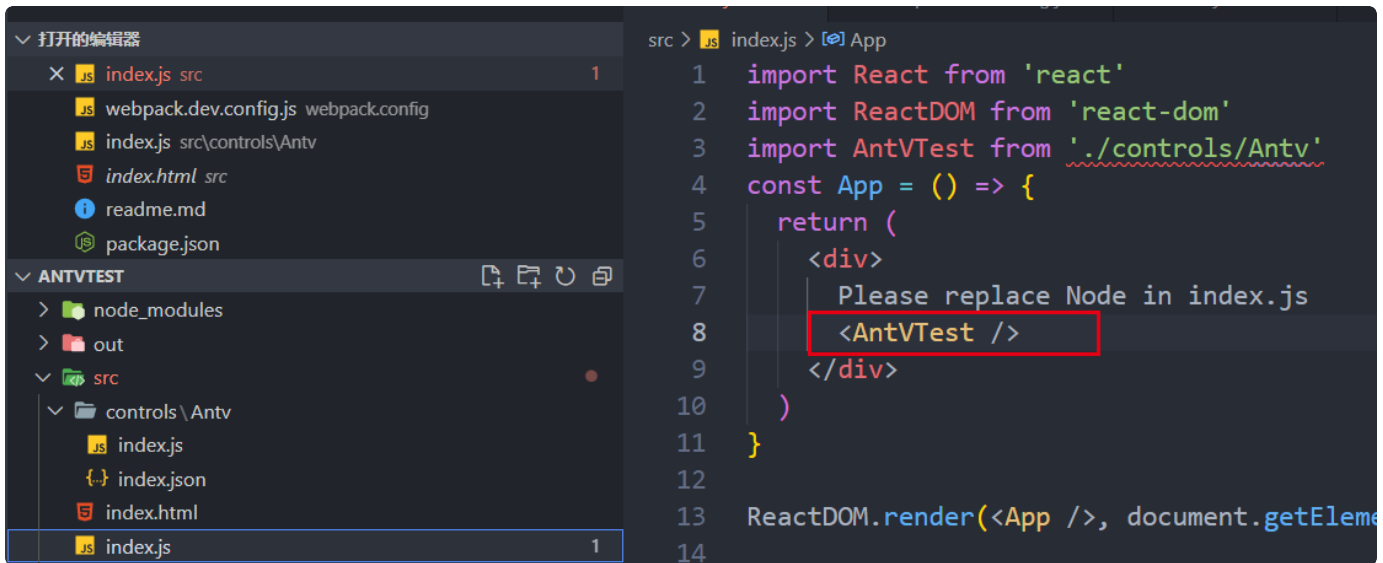
可编程组件开发

1. 在低代码app中引入一个可编程组件，假设命名为Antv
2. 在vscode 执行sync（在 VSCode 中快捷键〔ctrl+shift+p〕打开输入命令窗口，输入命令“sync”（选择 SVE Sync）），成功后会在src/controls下看到Antv文件夹，里面有index.js
3. 执行npm install @antv/l7 @antv/l7-map，引入antv/l7库
4. 正常开发Antv组件，注意引入方式为正常的"@antv/l7"



```
src > controls > Antv > index.js > AntVTest > useEffect() callback > scene > map
1 import React, { useEffect } from 'react'
2 import { Scene, PointLayer } from '@antv/l7'
3 import { Mapbox, GaodeMap } from '@antv/l7-maps'
4 const AntVTest = () => {
5   useEffect(() => {
6     const scene = new Scene({
7       id: 'antv-map',
8       map: new Mapbox({
9         pitch: 0,
10        style: 'dark',
11        center: [20, -3.69],
12        zoom: 2.5,
```

5. 在src/index.js中引入开发的Antv组件，并运行npm run start调试

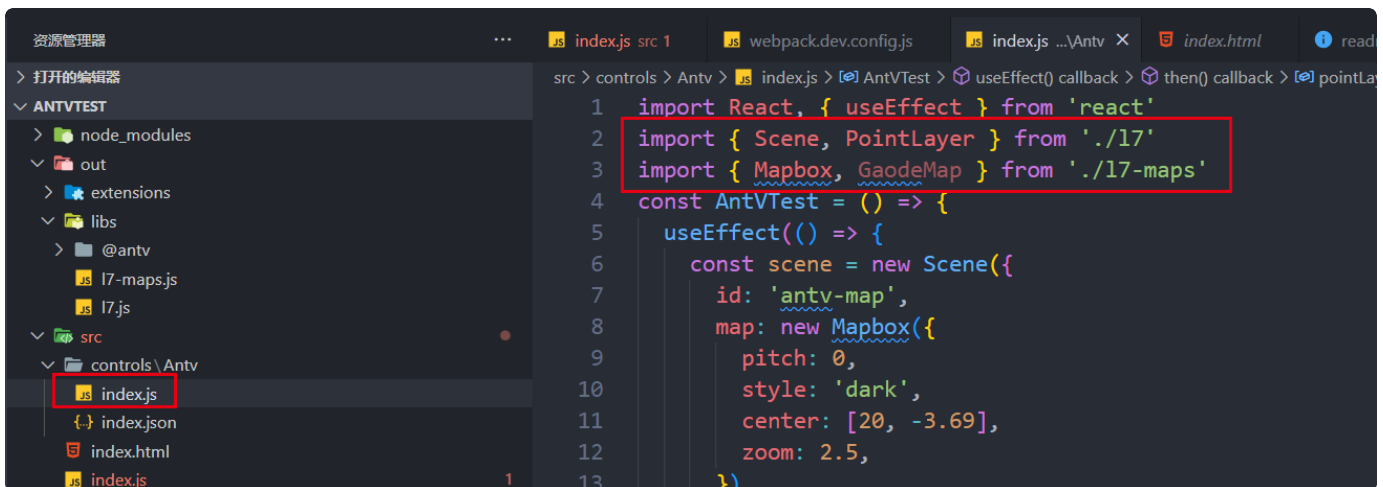


```
src > index.js > App
1 import React from 'react'
2 import ReactDOM from 'react-dom'
3 import AntVTest from './controls/Antv'
4 const App = () => {
5   return (
6     <div>
7       Please replace Node in index.js
8       <AntVTest />
9     </div>
10   )
11 }
12
13 ReactDOM.render(<App />, document.getEleme
14
```

6. 本地开发完成后，执行npm run build:lib，在out/libs目录下多出@antv文件夹

7. 将@antv文件夹下的js文件移到out/libs目录下，并将@antv文件夹删除

8. 修改引入方式



```
src > controls > Antv > index.js > AntVTest > useEffect() callback > then() callback > pointLa
1 import React, { useEffect } from 'react'
2 import { Scene, PointLayer } from './l7'
3 import { Mapbox, GaodeMap } from './l7-maps'
4 const AntVTest = () => {
5   useEffect(() => {
6     const scene = new Scene({
7       id: 'antv-map',
8       map: new Mapbox({
9         pitch: 0,
10        style: 'dark',
11        center: [20, -3.69],
12        zoom: 2.5,
```

9. 在vscode SVE:publish:all，将调试好的代码推到低代码app

10. 在低代码app中预览查看

