

3.1.x

项目说明

项目名	说明
jnpf-web	主项目前端
jnpf-web-datascreeen	大屏前端
jnpf-web-dataareport	报表前端
jnpf-web-tenant	多租户前端

环境要求

基础要求

因环境不同，在某些情况下无法顺利安装前端依赖，需要安装 Node.js 最新LTS版本及 git
本文档将以 最新LTS版本 进行演示

- Git
- Node.js
- Yarn

推荐IDE

- Visual Studio Code
- WebStorm

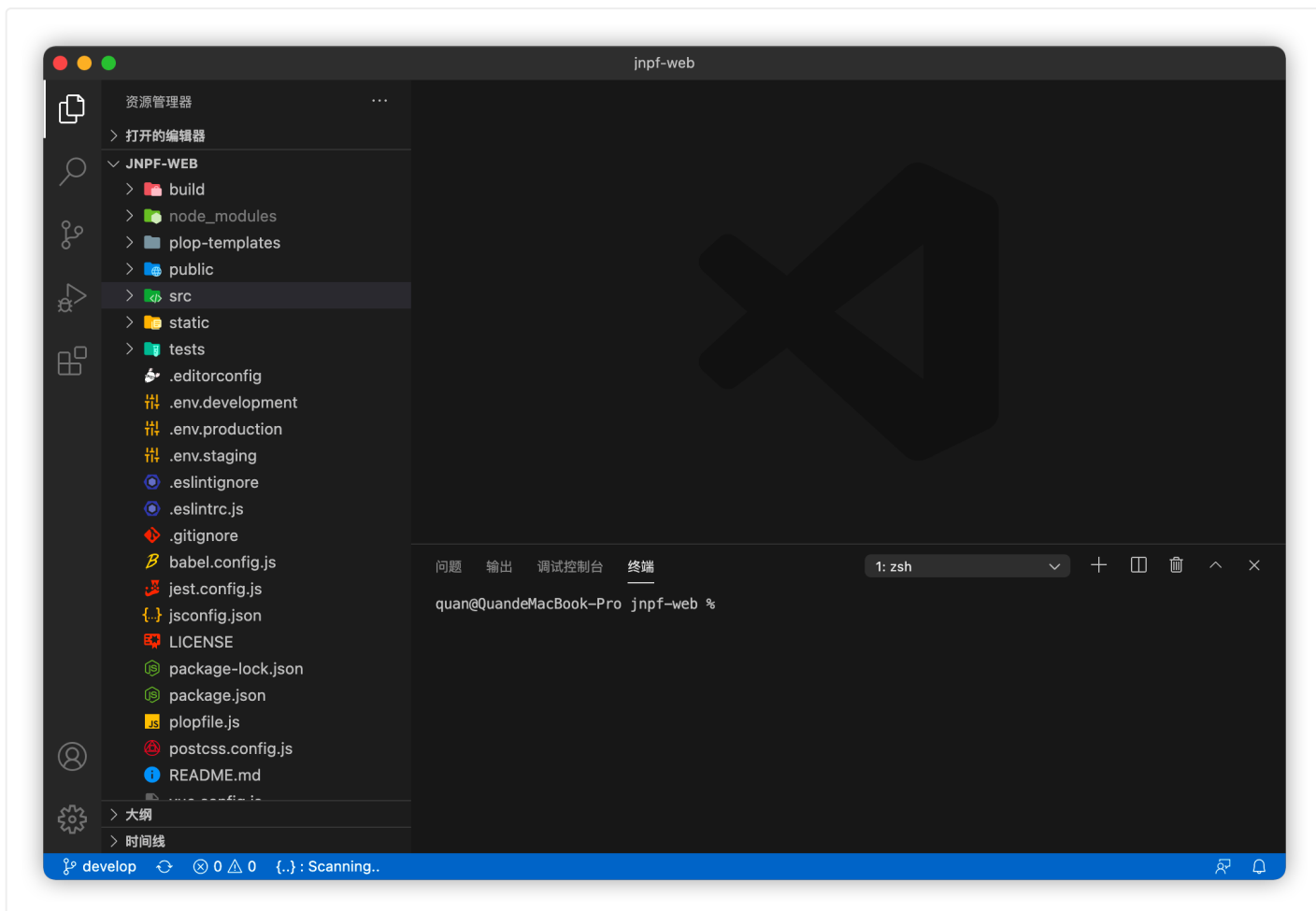
Visual Studio Code推荐插件

- Vetur
- Auto Close Tag
- Auto Rename Tag
- JavaScript(ES6) code snippets
- Beautify

项目导入

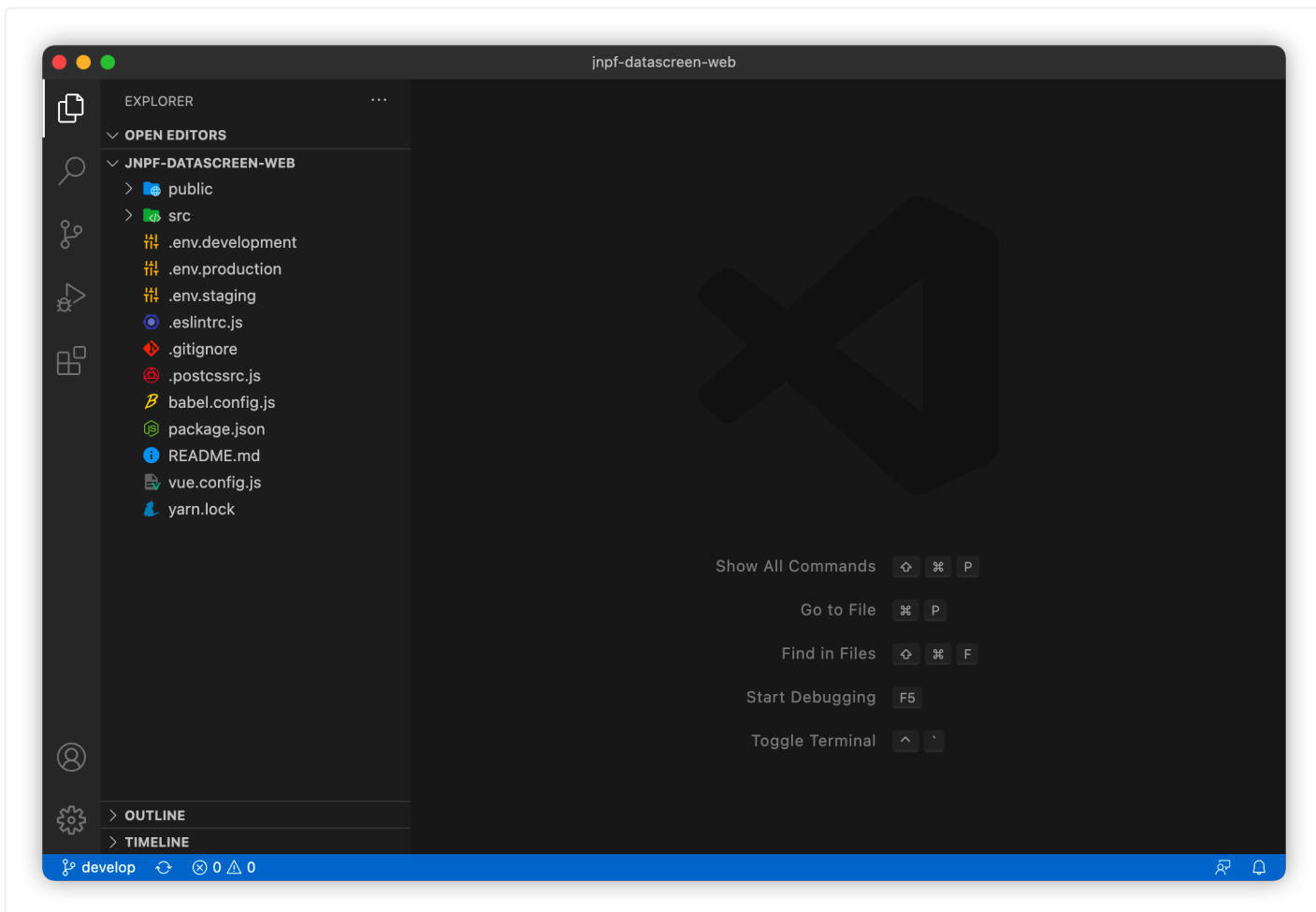
主项目前端

- 方式1: 打开 Visual Studio Code 编辑器, 把 jnpf-web 项目拖到编辑器主界面即可
- 方式2: 依次选择菜单 文件 -> 打开文件夹 , 选中 jnpf-web 前端工程目录



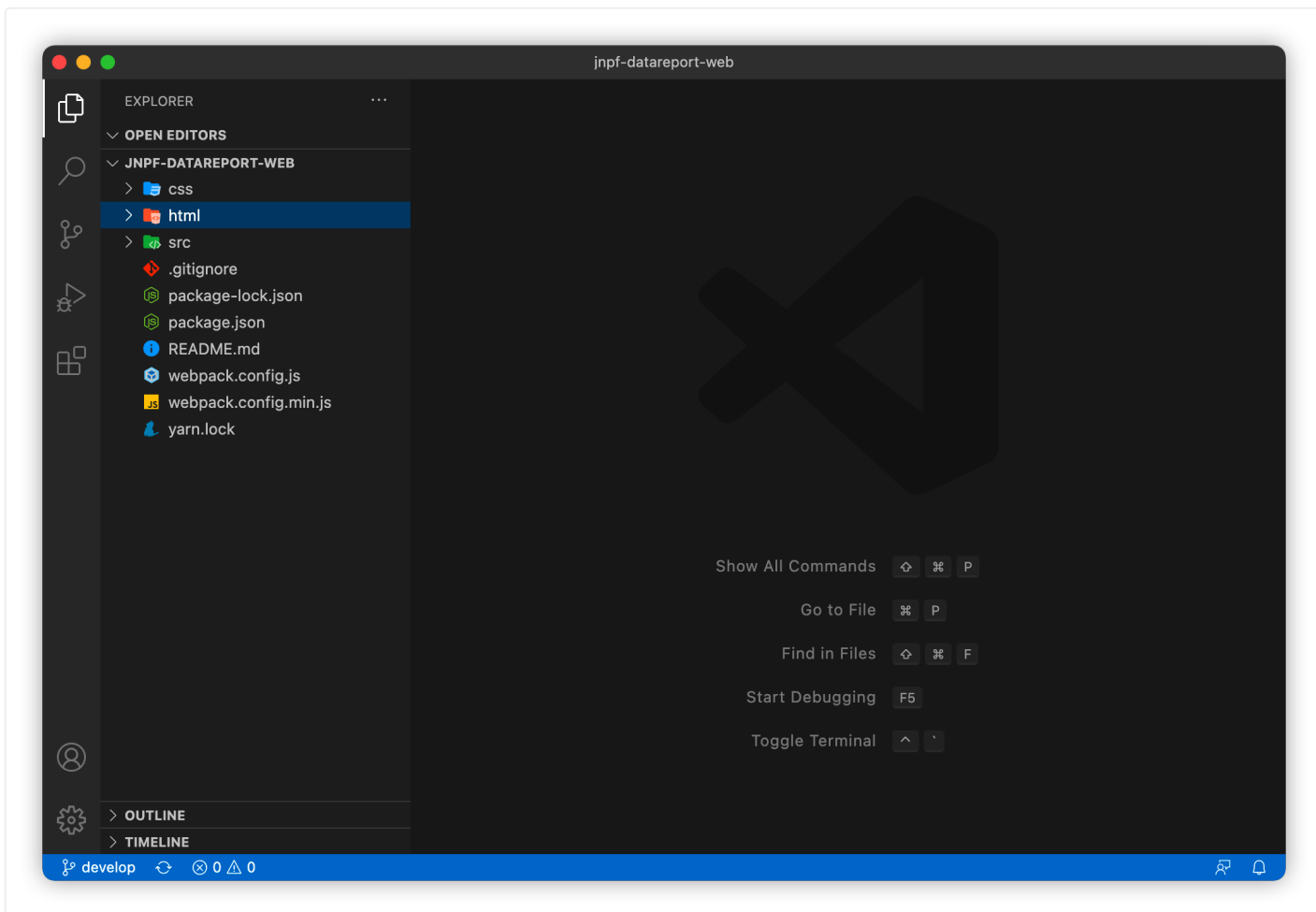
大屏前端

- 方式1: 打开 Visual Studio Code 编辑器, 把 jnpf-web-datascreen 项目拖到编辑器主界面即可
- 方式2: 依次选择菜单 文件 -> 打开文件夹 , 选中 jnpf-web-datascreen 前端工程目录



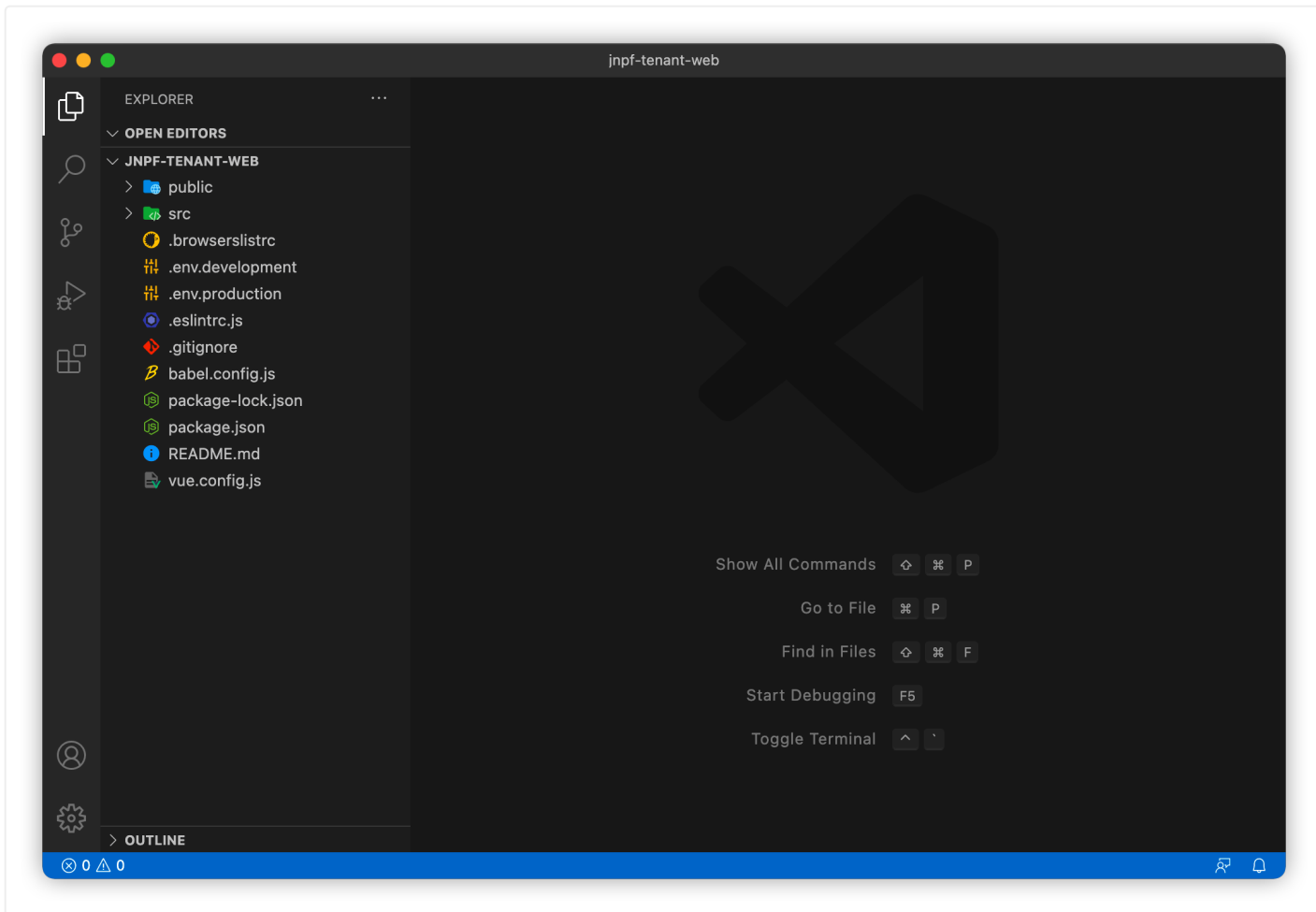
报表前端

- 方式1: 打开 Visual Studio Code 编辑器, 把 jnpf-web-datareport 项目拖到编辑器主界面即可
- 方式2: 依次选择菜单 文件 -> 打开文件夹 , 选中 jnpf-web-datareport 前端工程目录



多租户前端

- 方式1: 打开 Visual Studio Code 编辑器, 把 jnpf-web-tenant 项目拖到编辑器主界面即可
- 方式2: 依次选择菜单 文件 -> 打开文件夹 , 选中 jnpf-web-tenant 前端工程目录



项目运行

主项目前端

1) 打开 `src/utils/define.js` ,修改开发环境接口

```
// 开发环境接口配置
const APIURL = 'http://192.168.0.25:30000'

module.exports = {
  APIURL: APIURL,
  timeout: process.env.NODE_ENV === 'development' ? 10000 : 1000000,
  WebSocketUrl: process.env.NODE_ENV === 'development' ? APIURL.replace('http', 'ws') + '/websocket' : process.env.VUE_APP_BASE_WSS,
  comUploadUrl: process.env.VUE_APP_BASE_API + '/api/file/Uploader',
  comUrl: process.env.VUE_APP_BASE_API,
  // 大屏应用前端路径
  dataV: process.env.NODE_ENV === 'development' ? 'http://localhost:8100/DataV' : process.env.VUE_APP_BASE_API + '/DataV',
  // 数据报表
  reportServer: process.env.NODE_ENV === 'development' ? 'http://localhost:9000' : p
```

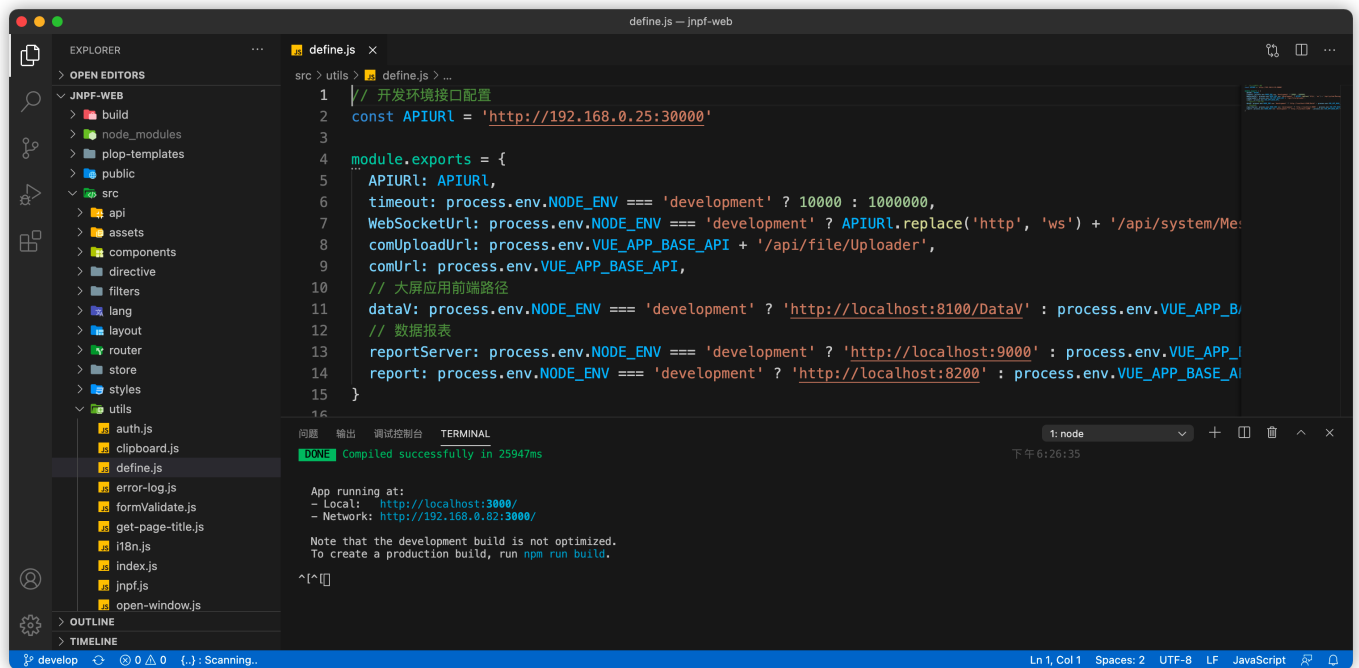
```
process.env.VUE_APP_BASE_API + '/ReportServer',
  report: process.env.NODE_ENV === 'development' ? 'http://localhost:8200' : process.env.VUE_APP_BASE_API + '/Report'
}
```

2) 依次选择菜单 终端 -> 新终端，执行以下命令，安装项目所需依赖

```
npm i
```

3) 安装完依赖后，执行以下命令来运行您的项目，

```
npm run dev
```



3) 运行成功后，会自动打开项目界面，如下图所示



大屏前端

- 1) 打开 `src/utils/define.js` ,修改开发环境接口

```
// 开发环境接口配置
const APIURL = 'http://192.168.0.25:30000'

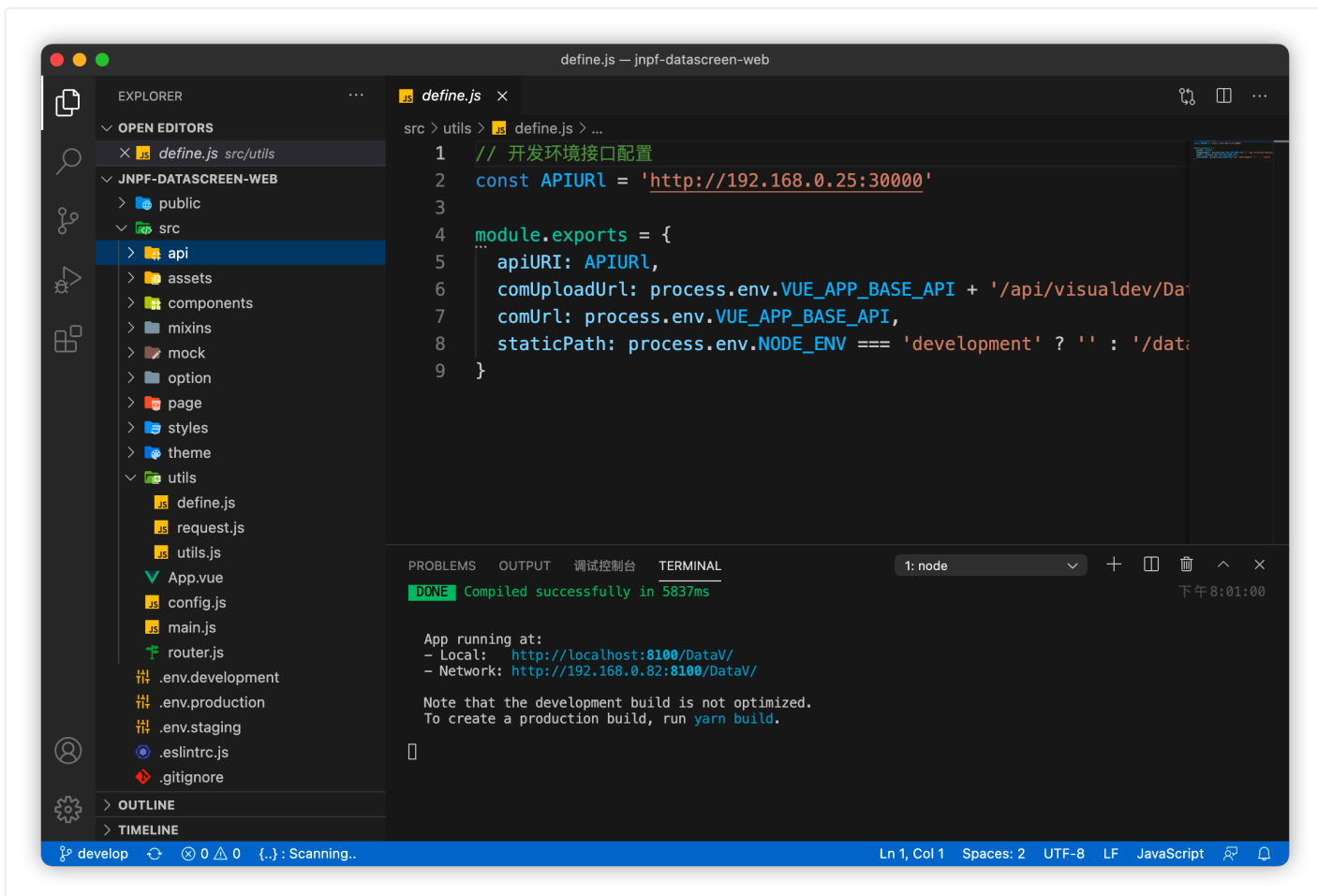
module.exports = {
  apiURI: APIURL,
  comUploadUrl: process.env.VUE_APP_BASE_API + '/api/visualdev/DataScreen/Images/',
  comUrl: process.env.VUE_APP_BASE_API,
  staticPath: process.env.NODE_ENV === 'development' ? '' : '/dataV'
}
```

- 2) 依次选择菜单 终端 -> 新终端 , 执行以下命令, 安装项目所需依赖

```
yarn
```

- 3) 安装完依赖后, 执行以下命令来运行您的项目

```
yarn dev
```



3) 运行成功后，通过主项目前端界面打开，【在线开发】 - 【大屏设计】 - 【新建大屏】中查看（建议先运行后端）

报表前端

1) 安装 http-server

```
npm install --global http-server
```

2) 进入 html 目录，并修改 index.html 和 preview.html 的配置

index.html文件

```
<script type="text/javascript">
  // 报表接口
  window._server = "http://localhost:9000";
  // 报表前端
  window._contextPath = "http://localhost:8200";
  // 主项目接口地址
  window._mainServer = "http://127.0.0.1:7772";
</script>
```

preview.html


```
<script type="text/javascript">
  // 报表接口
  window._server = "http://localhost:9000";
  window.formElements = [];
</script>
```

3) 依次选择菜单 终端 -> 新终端 ，执行以下命令

```
http-server -a localhost -p 8200
```

4) 运行成功后，通过主项目前端界面打开，【在线开发】 - 【报表设计】 - 【新建报表】 中查看（建议先运行后端）

多租户前端

1) 打开 src/utils/define.js ,修改开发环境接口

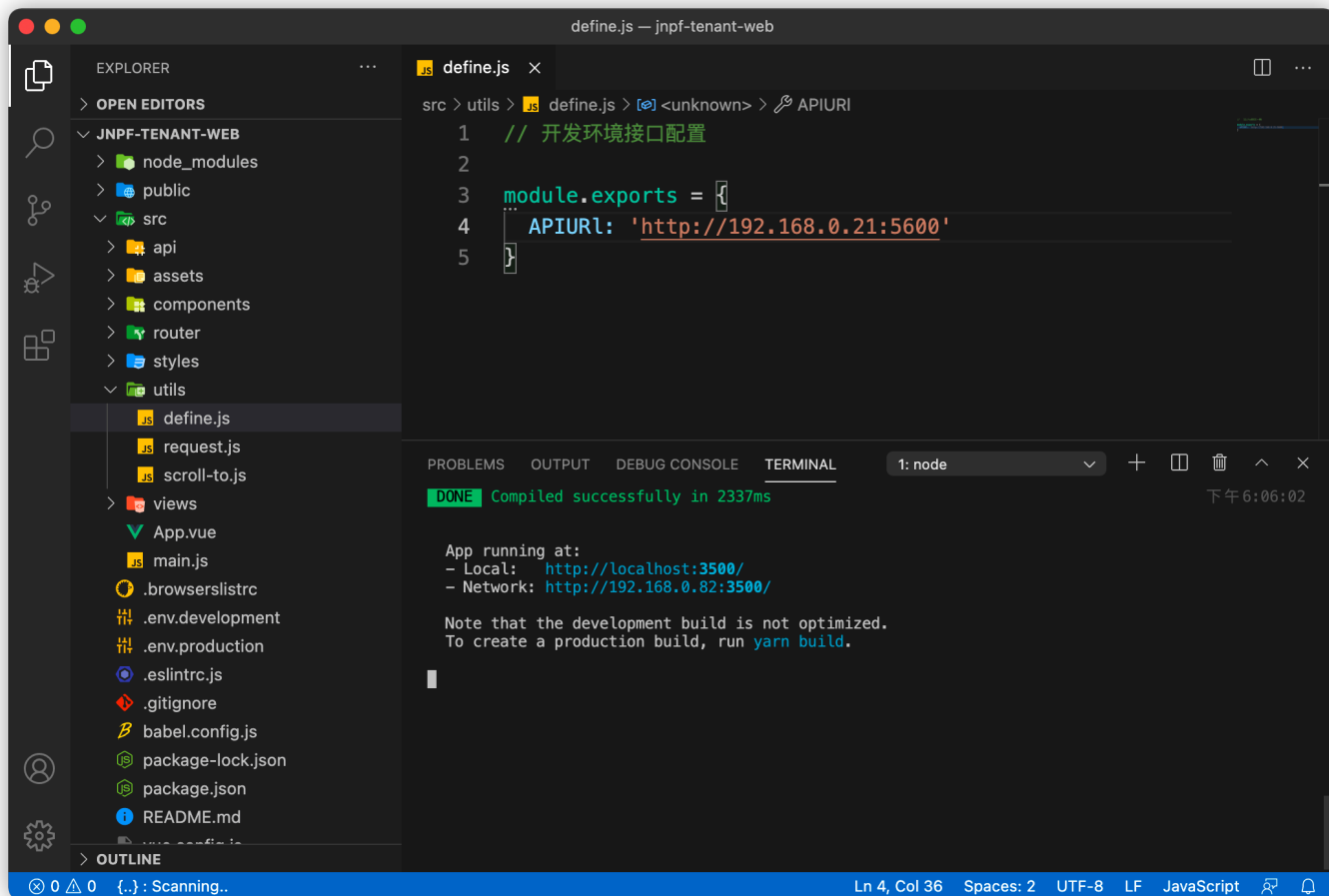
```
// 开发环境接口配置
module.exports = {
  APIURL: 'http://192.168.0.100:7100'
}
```

2) 依次选择菜单 终端 -> 新终端 ，执行以下命令，安装项目所需依赖

```
yarn
```

3) 安装完依赖后，执行以下命令来运行您的项目

```
yarn dev
```



3) 运行成功后，会自动打开项目界面，如下图所示

