

# 可视化传染病知识图谱可视化介绍

## 一、项目运行

### 1. 依赖安装

```
# 二选一
npm i
# yarn需要全局安装一下
yarn install
```

(需要提前配置好npm或yarn)

使用 `cnpm` (国内淘宝镜像) 来安装, 速度更快:

```
# 二选一
npm install -g cnpm --registry=https://registry.npm.taobao.org
npm install cnpm -g --registry=https://registry.nlark.com
```

```
cnpm i
```

### 2. 项目启动

```
# 若启动前报错, 运行下行命令降低node.js版本
set NODE_OPTIONS=--openssl-legacy-provider

# 二选一
npm run serve
yarn serve
```

### 3. 项目打包

```
# 二选一
npm run build
yarn build
```

## 二、代码说明

### 页面结构

页面通过组件分为不同功能模块:

1. `2dview.vue` — 2D图谱展示页面
  - `d3graph.vue`
    - 2D图谱展示组件
  - `gSearch.vue`
    - 搜索组件

- 目前模拟后台请求，通过require代替

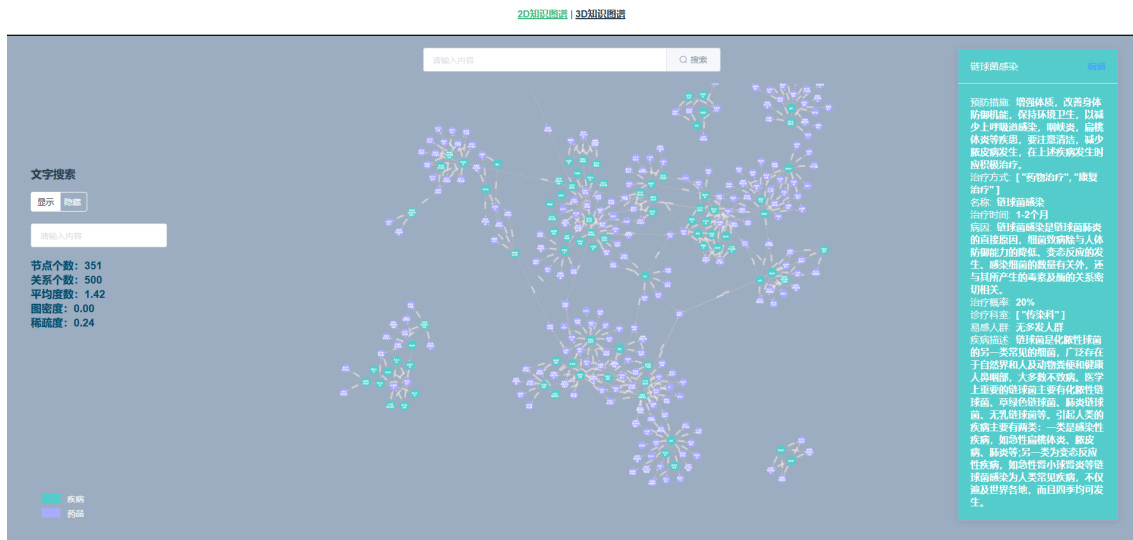
## 2. 3dview.vue — 3D图谱展示页面

- threeGraph.vue
  - 3D图谱展示组件
- gSearch.vue
  - 搜索组件
  - 目前模拟后台请求，通过require代替

# 三、代码效果展示

分为2D、3D图谱展示两个页面：

## 1. 2D图谱展示



## 2. 3D图谱展示



# 四、部署项目在github服务器

- 1 配置vue.config.js

在当前目录下配置 vue.config.js 如下，若没有 vue.config.js 则手动创建

```
//vue.config.js
const port = 3000; // 端口配置
```

```
module.exports = {
  devServer: {
    port, //设置端口
  },
  publicPath: './', //修改目录
  outputDir: 'dist', // build时构建文件的目录 构建时传入 --no-clean 可关闭该行为
  assetsDir: 'assets', // build时放置生成的静态资源 (js、css、img、fonts) 的 (相对于
outputDir 的) 目录
  filenameHashing: true, // 默认在生成的静态资源文件名中包含hash以控制缓存
  runtimeCompiler: false, // 是否使用包含运行时编译器的 vue 构建版本
  productionSourceMap: true, // 如果你不需要生产环境的 source map, 可以将其设置为
false 以加速生产环境构建
};
```

- 2 项目打包

在当前目录打包项目

```
# 二选一
npm run build
yarn build
```

生成 `./dist/` 文件夹, 点击 `dist/index.html`, 若能正常加载页面, 即可直接上传至github仓库。若加载出为空白, 需要更改 `./vue.config.js`

参考: [https://blog.csdn.net/weixin\\_48337566/article/details/123775830](https://blog.csdn.net/weixin_48337566/article/details/123775830)

按上述链接内容修改js文件时, 注意端口号设置。

修改后, 重新build打包文件, 生成 `./dist` 文件夹

- 3 上传github仓库

参考: <http://www.45fan.com/article.php?aid=1D2EkkiFXF6yg3nA>

直接dist文件夹上传仓库, 并按上述链接生成网页即可