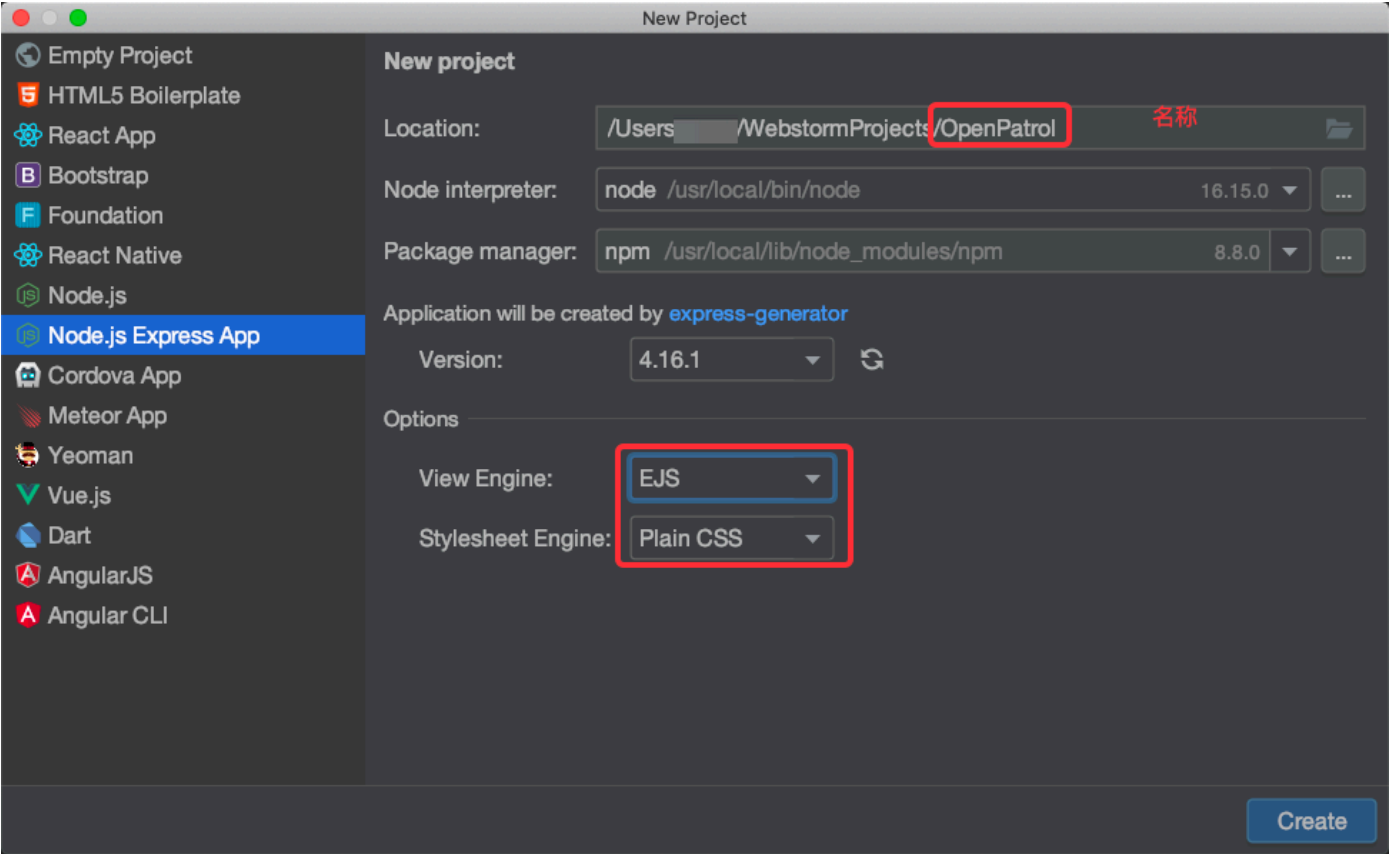


本文主要介绍创建工程，目录设置，代码仓库上传

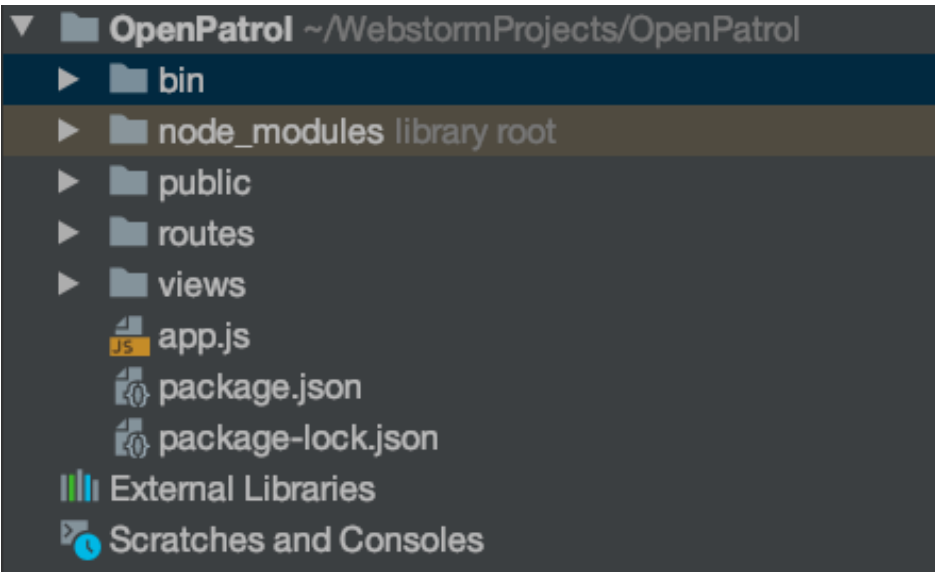
创建工程

1-Webstorm创建Node.js工程

路径：Webstorm->File->New->Project

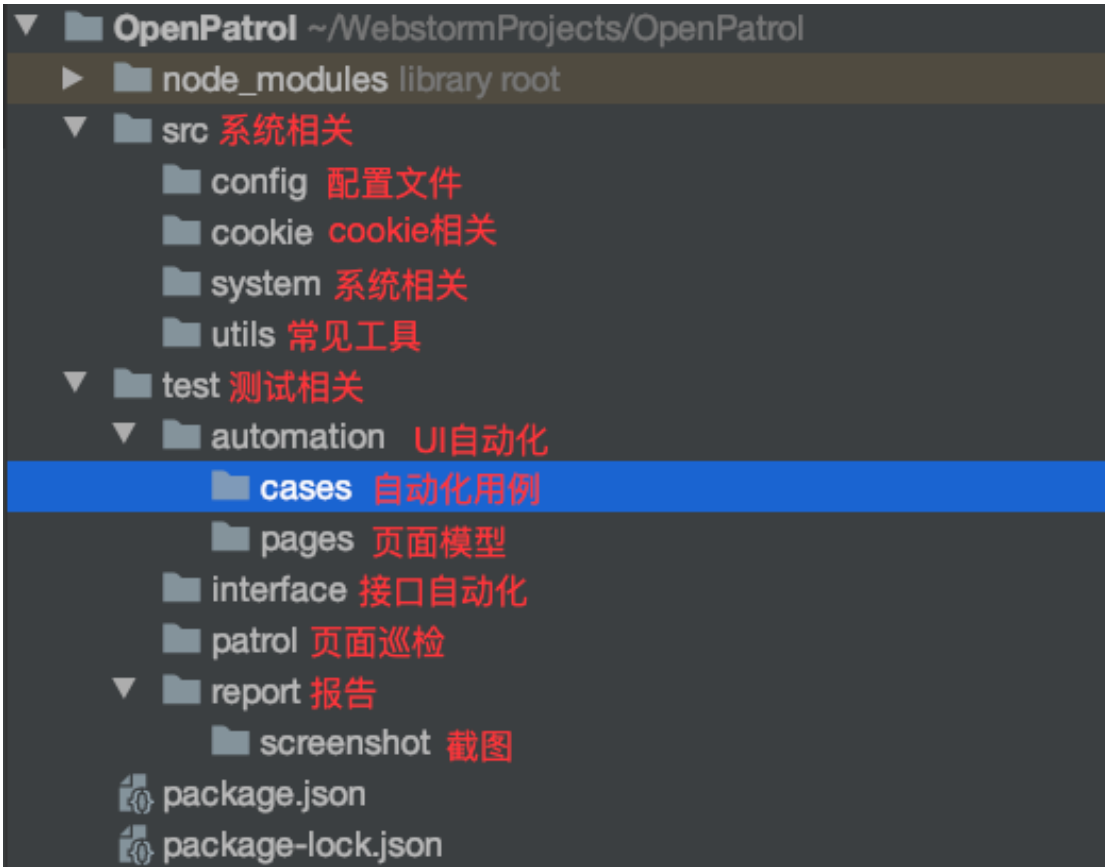


默认的工程结构：

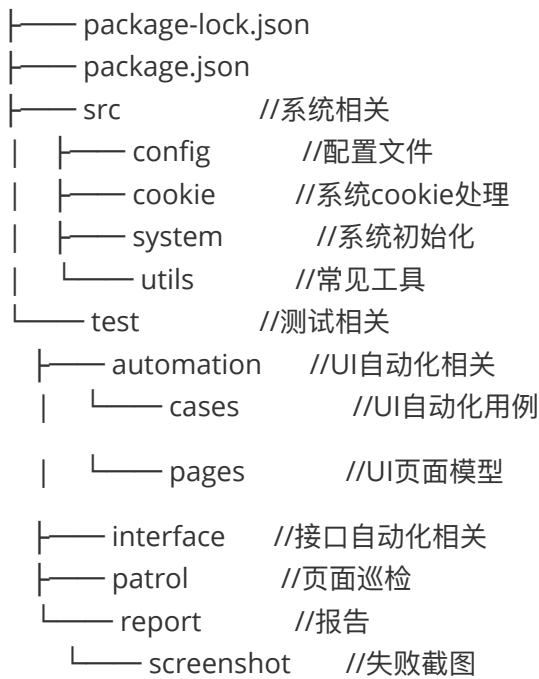


2-工程目录设置

删除不需要的目录，初始化目录参考如下（可以按照个人习惯设置，这里仅参考）， 后续开发过程中按需设置



Tree结构:

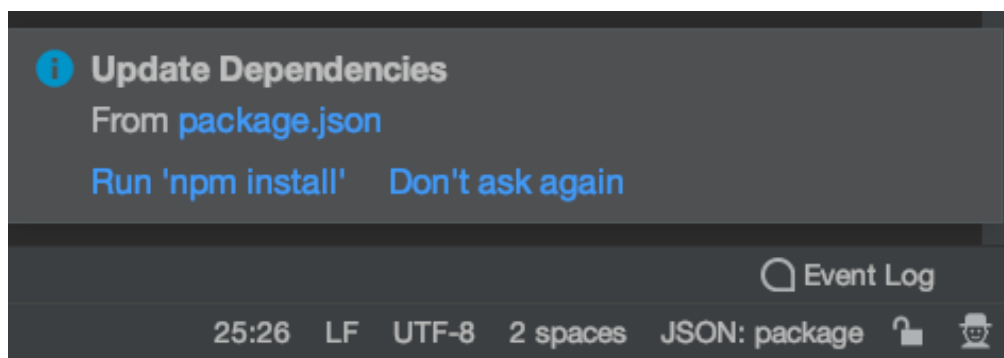


3-package.json初始化

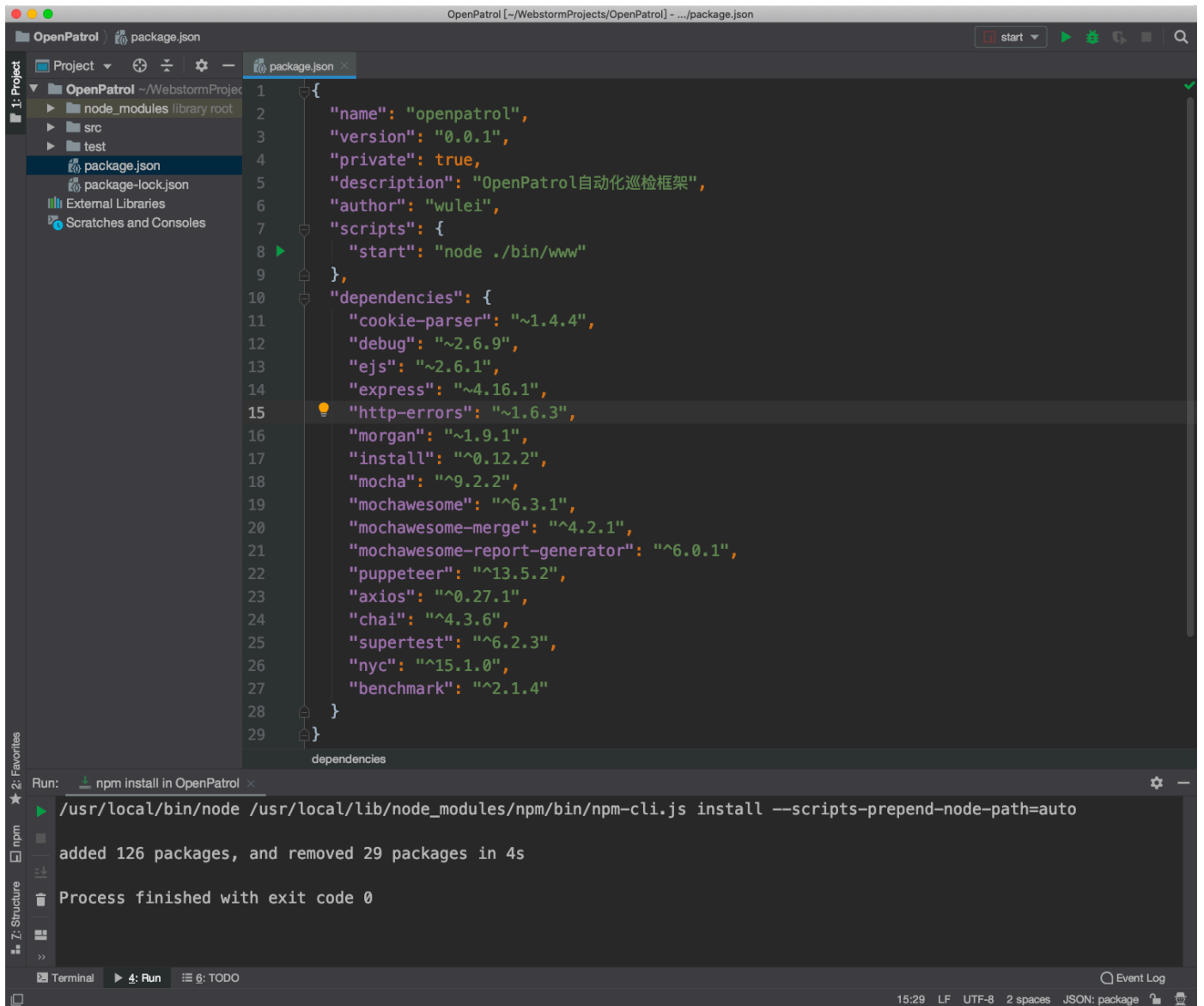
1. package.json配置相关的依赖

```
{
  "name": "openpatrol",
  "version": "0.0.1",
  "private": true,
  "description": "OpenPatrol自动化巡检框架",
  "author": "wulei",
  "scripts": {
    "start": "node ./bin/www"
  },
  "dependencies": {
    "cookie-parser": "~1.4.4",
    "debug": "~2.6.9",
    "ejs": "~2.6.1",
    "express": "~4.16.1",
    "http-errors": "~1.6.3",
    "morgan": "~1.9.1",
    "install": "^0.12.2",
    "mocha": "^9.2.2",
    "mochawesome": "^6.3.1",
    "mochawesome-merge": "^4.2.1",
    "mochawesome-report-generator": "^6.0.1",
    "puppeteer": "^13.5.2",
    "axios": "^0.27.1",
    "chai": "^4.3.6",
    "supertest": "^6.2.3",
    "nyc": "^15.1.0",
    "benchmark": "^2.1.4"
  }
}
```

2. webstorm右下角会提示进行安装依赖

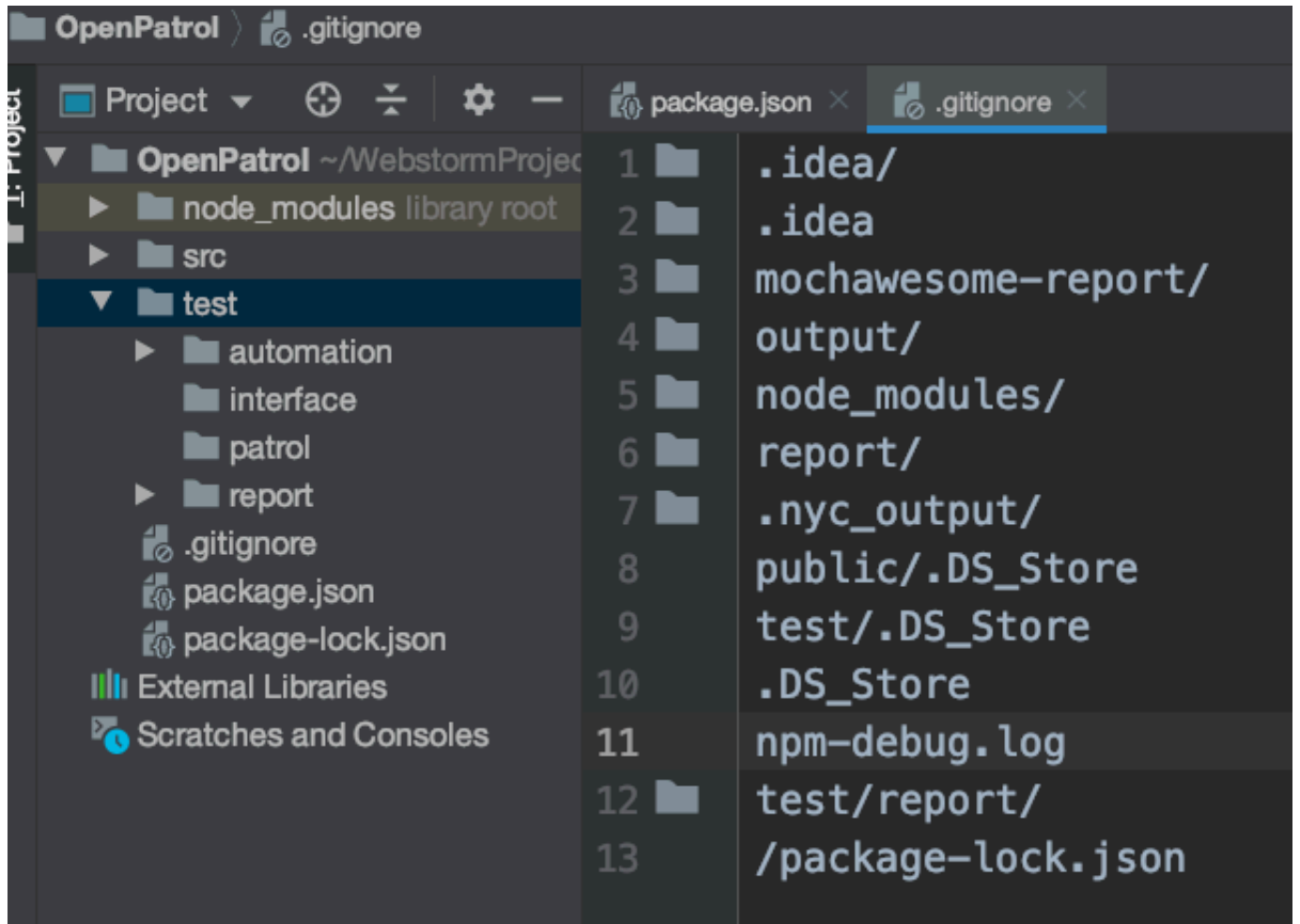


3. 安装完成



4-设置ignore文件

在工程目录新建一个.gitignore文件



文件内容：

```
.idea/  
.idea  
mochawesome-report/  
output/  
node_modules/  
report/  
.nyc_output/  
public/.DS_Store  
test/.DS_Store  
.DS_Store  
npm-debug.log  
test/report/  
/package-lock.json
```

5-代码仓库创建

如果是个人在公司Gitlab或者采用开源的GitHub，公司情况下可以联系对应的管理员创建空的代码仓库。

创建后的仓库例子： git@gitlab.***.com:test/openpatrol.git

6-本地代码上传仓库

推送本地目录到远程仓库流程：

```
#cd OpenPatrol
#git init
#git remote add origin git@gitlab.**.com:qa/openpatrol.git
#git add .
#git commit -m "Initial commit"
#git push -u origin master
```

此时查看远程仓库，只有2个文件，原因是因为初始化的目录都是空，等后续有文件后再次上传即可

master

openpatrol / +

History

Find file

Web IDE

Clone

Initial commit
[redacted] authored 1 minute ago

e9893c11

Name	Last commit	Last update
.gitignore	Initial commit	1 minute ago
package.json	Initial commit	1 minute ago