

package.json的正确使用姿势

📅 发表于 2017-12-29 | 🔄 更新于 2020-01-02 | 📁 Node

📄 字数总计: 1k | ⌚ 阅读时长: 3 分钟 | 👁 阅读量: 3172

写在前面

刀耕火种的时代已经过去，前端开发急需自动化，工程化。

有一天，你找到公司新人程序员小T和小F，给他们每个人分配了一个任务。小T和小F微笑的答应了，确定了项目的选型和结构，小T和小F准备安装依赖的模块。但是他们开始对于dependencies和devDependencies并没有那么在意：

程序员小T把所有的依赖模块都使用 `npm install --save`，将依赖安装在项目中，并写入了package.json的 `dependencies` (生产环境)，最终导致项目臃肿不堪，前端是对性能的狂热追求者，最终小T的项目没有通过测试。

程序员小F把所有的依赖模块使用 `npm install --save-dev`，将依赖安装在项目中，并写入了package.json中的 `devDependencies` (本地开发环境)，最终导致正常运行该项目使用了哪些依赖模块无从得知，预想后期会带来很多麻烦，导致小F的项目也没有通过测试

安装依赖的三种方式

小T和小F有些疑惑，项目本地明明正常，为什么都没有通过测试呢？他们决定弄个明白..

于是小T问，小F“你是怎么安装依赖的呀？”小F：“我是使用 `npm install --save-dev` 安装依赖模块”，“哦，是这样呀，我是用 `npm install --save` 安装的模块”，小T答到。

- `npm install`
- `npm install --save`
- `npm install --save-dev`

使用 `npm install` 将依赖模块安装到项目，但不写 `package.json`（不推荐，这样最终需要手动添加）；使用 `npm install --save` 将模块安装到项目，写入的 `package.json` 的 `dependencies` 中；使用 `npm install --save-dev` 将模块安装到本地，写入 `package.json` 的 `devDependencies` 中。

```
"dependencies": {
  "body-parser": "~1.15.2",
  "cookie-parser": "~1.4.3",
  "debug": "~2.2.0",
  "ejs": "~2.5.2",
  "express": "~4.14.0",
  "express-session": "^1.14.2",
  "morgan": "~1.7.0",
  "serve-favicon": "~2.3.0"
},
"devDependencies": {
  "express-session": "^1.14.2",
  "supervisor": "^0.12.0"
}
```

开发环境生产环境

小T恍然大悟，原来是这样呀。那么前端工程化中的 `开发环境` 和 `生产环境` 到底有怎么样的区别呢？

正确使用姿势

那么我们如何知道哪些是在生产环境中需要，哪些是在开发环境中需要呢？

生产环境只需要我们能正常运行该项目所需要的模块，比如vue，vue-router，express，jQuery等等这些，项目没有这些依赖会出错

开发环境就是开发阶段，我们所做的单元测试，webpack，gulp，supervisor等这些工具，都只是在开发阶段需要，一旦项目投入需要便不再需要。

使用npm install 默认安装dependencies和devDependencies中的模块，如果只需要安装生产环境中的模块使用npm install -production

区别

原来devDependencies里面的模块只用于开发环境，不用于生产环境，而 dependencies 是需要发布到生产环境的。那究竟有什么区别呢？工程化为什么需要这两个环境？

开发环境

devDependencies中dev是develop的缩写，表示开发环境，即指开发阶段，对于一些仅在开发阶段需要的模块，项目正式投入后便不再需要的模块,我们选择把它们安装在

`devDependencies`。

前端开发越来越体现工程化改革的姿态，无论是使用工具的进化，还是框架的更新迭代，前端对性能狂热追求伴随着工程化的进步也在不断提高，必然导致开发环境的多种多样，不可避免的导致开发环境的臃肿

生产环境

生产环境也就是真是真实环境,是线上用户接触的产环境,因为开发环境的臃肿，不能直接用于生产环境，我们需要对环境可以优化的部分进行优化。

判断是开发环境还是生产环境

- 使用req.app.get('env')

举个栗子：

▼ javascript

```
app.use(function(err, req, res, next) {  
  
  // set locals, only providing error in development
```

```
res.locals.message = err.message;

res.locals.error = req.app.get('env') === 'development' ? err : {};

// render the error page

res.status(err.status || 500);

res.render('error');

});
```

```
Press rs for restarting the process.
development
GET / 304 17.509 ms - -
```

req.app.get('env')

参考资料

[qdfuns notes](#)

[npm package.json](#)

文章作者: [okaychen](#)



版权声明: 本博客所有文章除特别声明外，均采用 [CC BY-NC-SA 4.0](#) 许可协议。转载请注明来自 [前端随笔](#)！

Node

