

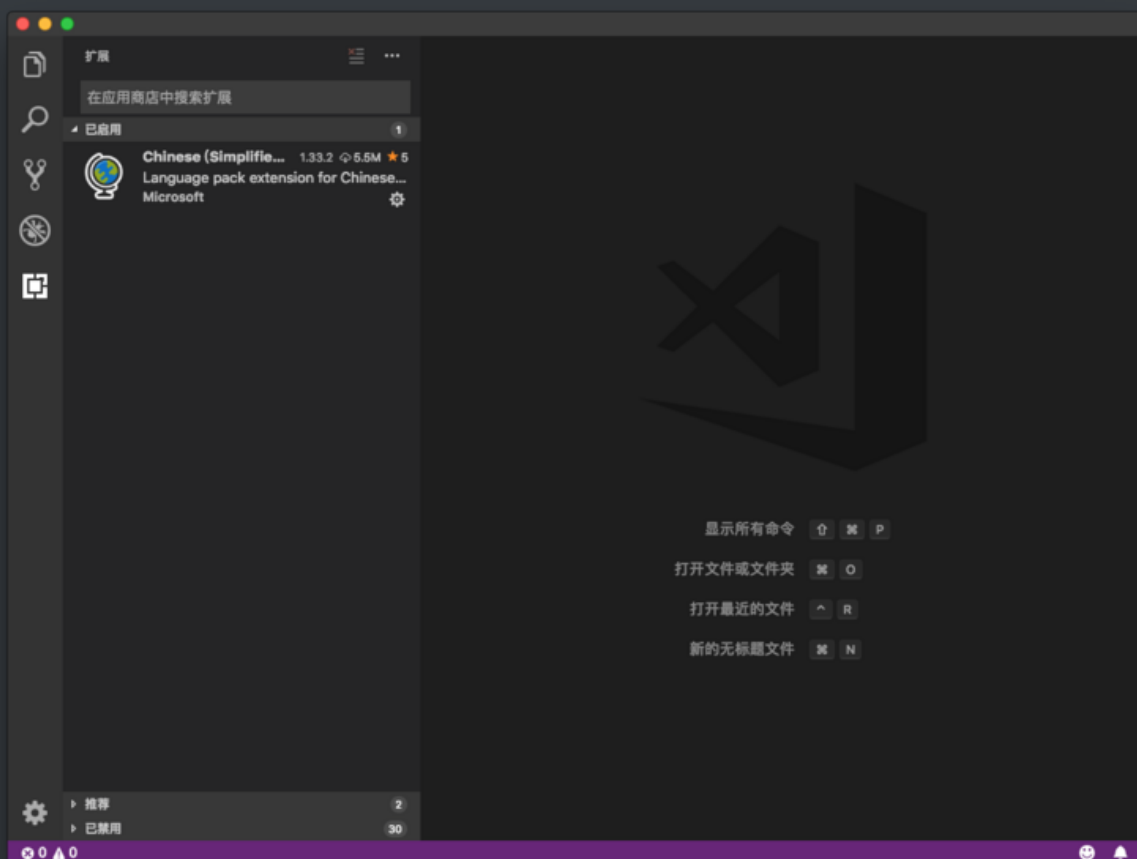
用vscode开发vue应用

2019-05-02

原文链接: segmentfault.com

从安装开始

为了准确起见，我们把 vscode 里所有插件全部禁用，把用户设置清空，以让它尽可能恢复成原始的样子：



作为代码洁癖患者，对于系统的版本要求一定也是最苛刻的，不管什么时候，都让我们把所有的系统能升级的都升级到最高版本：

```
npm install -g @vue/cli
```

然后，我们开始创建项目：

```
vue create hello-world
```

在这里，一定要选择第一项：babel + eslint，这两个是必不可少的。我见到有些人嫌 eslint 麻烦，居然在项目建立好之后手工把 eslint 关掉的，简直无语。

```
Vue CLI v3.7.0
? Please pick a preset: (Use arrow keys)
> default (babel, eslint)
  Manually select features
```

安装完毕：

```
Vue CLI v3.7.0
🌟 Creating project in /Users/zhangjing/Projects/hello-world.
📁 Initializing git repository...
⦿ Installing CLI plugins. This might take a while...

yarn install v1.15.2
info No lockfile found.
[1/4] 🔍 Resolving packages...
[2/4] 📦 Fetching packages...
[3/4] 🔗 Linking dependencies...
[4/4] 🏗 Building fresh packages...
success Saved lockfile.
🌟 Done in 44.03s.
🔧 Invoking generators...
📦 Installing additional dependencies...

yarn install v1.15.2
[1/4] 🔍 Resolving packages...
[2/4] 📦 Fetching packages...
[3/4] 🔗 Linking dependencies...
[4/4] 🏗 Building fresh packages...

success Saved lockfile.
🌟 Done in 28.97s.
🔗 Running completion hooks...

📄 Generating README.md...

🎉 Successfully created project hello-world.
👉 Get started with the following commands:

$ cd hello-world
$ yarn serve
```

我们先不急着执行，执行代码是最容易的事情，我们先打开代码看一下：

好吧，至少我们需要先安装vetur插件。这几乎已经确定是开发 vue 项目的标配了，即使我不说，vscode 也会强烈建议你安装它。

装上 vetur 以后多少有点人样了。接下来我们来试一试能不能自动格式化，这部分才是洁癖患者的最爱。胡乱加几个空格，然后保存试试看

不能格式化，连个提示都没有！

用lint格式化

就算 vscode 里的 vetur 不能帮我们自动格式化，好在 package.json 命令里还有一个 lint 命令，我们看看 lint 命令能不能帮我们自动格式化：

lint 居然说没有错误！明明就是多了很多空格的错误好吧，为什么？

这是因为缺省的 vue-cli 没有为我们安装 @vue/prettier 插件，我们先来手工安装一下：

```
yarn add -D @vue/eslint-config-prettier
```

然后在 package.json 或者 .eslintrc.js 或者其它什么你设置 eslint 的地方，给它加上：

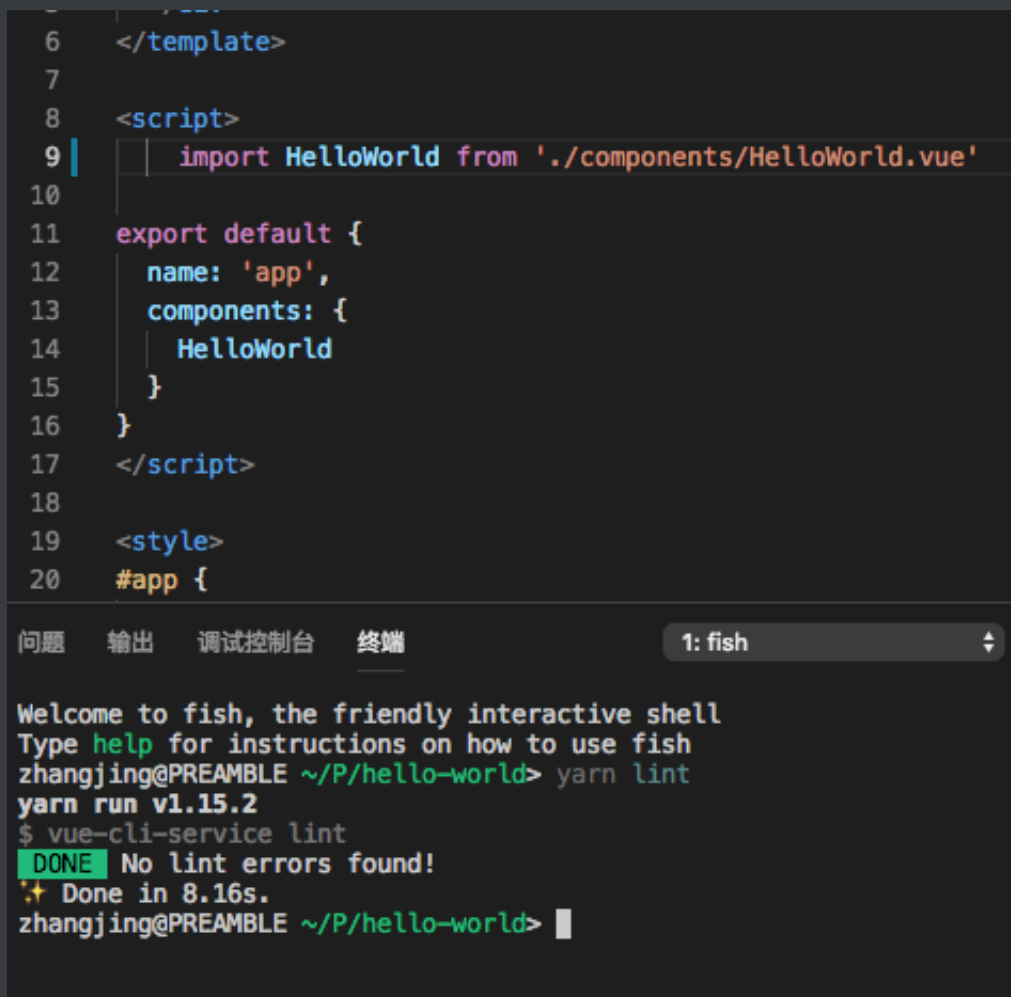
```
"extends": [  
  "plugin:vue/essential",  
  "eslint:recommended",  
  "@vue/prettier"  
],
```

特别是最后这一个 @vue/prettier，非常重要。然后再执行 yarn lint。多余的空格被自动干掉了，但是我们发现有一些地方同时也被篡改了

所有的单引号被变成双引号了，原本行尾没有的分号被加上了分号。这是为什么呢？因为我们虽然引入了 prettier，但是还没有对 prettier 做设置，我们在项目的根目录下创建一个 .prettierrc.js 文件，然后在其中加入：

```
module.exports = {  
  semi: false,  
  singleQuote: true  
}
```

再次执行 `yarn lint`，现在我们看到 `lint` 已经能够起作用了。它不但能把我们多余插入的空格删掉，并且能按照规则把双引号变成单引号，把行尾多余的分号删掉。当然，关于行尾加不加分号这是一个哲学命题，你可以根据你个人的喜好自行决定。在这里，我们权且按照 `vue-cli` 的标配执行。



```
6 </template>
7
8 <script>
9   import HelloWorld from './components/HelloWorld.vue'
10
11   export default {
12     name: 'app',
13     components: {
14       HelloWorld
15     }
16   }
17 </script>
18
19 <style>
20 #app {
```

问题 输出 调试控制台 终端 1: fish

```
Welcome to fish, the friendly interactive shell
Type help for instructions on how to use fish
zhangjing@PREAMBLE ~/P/hello-world> yarn lint
yarn run v1.15.2
$ vue-cli-service lint
DONE No lint errors found!
🌟 Done in 8.16s.
zhangjing@PREAMBLE ~/P/hello-world>
```

到这一步很关键，假设你拿到一个烂的不再烂的 `vue` 项目，里面有几千个 `.vue` 文件，几万个各种格式错误，也都能通过 `yarn lint` 这一行命令把它们全部修正过来！

在vscode里格式化

事情还没有完，我们注意到虽然 `yarn lint` 命令能在编写完代码之后帮我们格式化，但是既然我们是用 `vscode` 进行开发，我们当然希望能在 `vscode` 里直接看到对于错误的标注。

这时候我们需要在 `vscode` 里再安装一个插件 `eslint`。

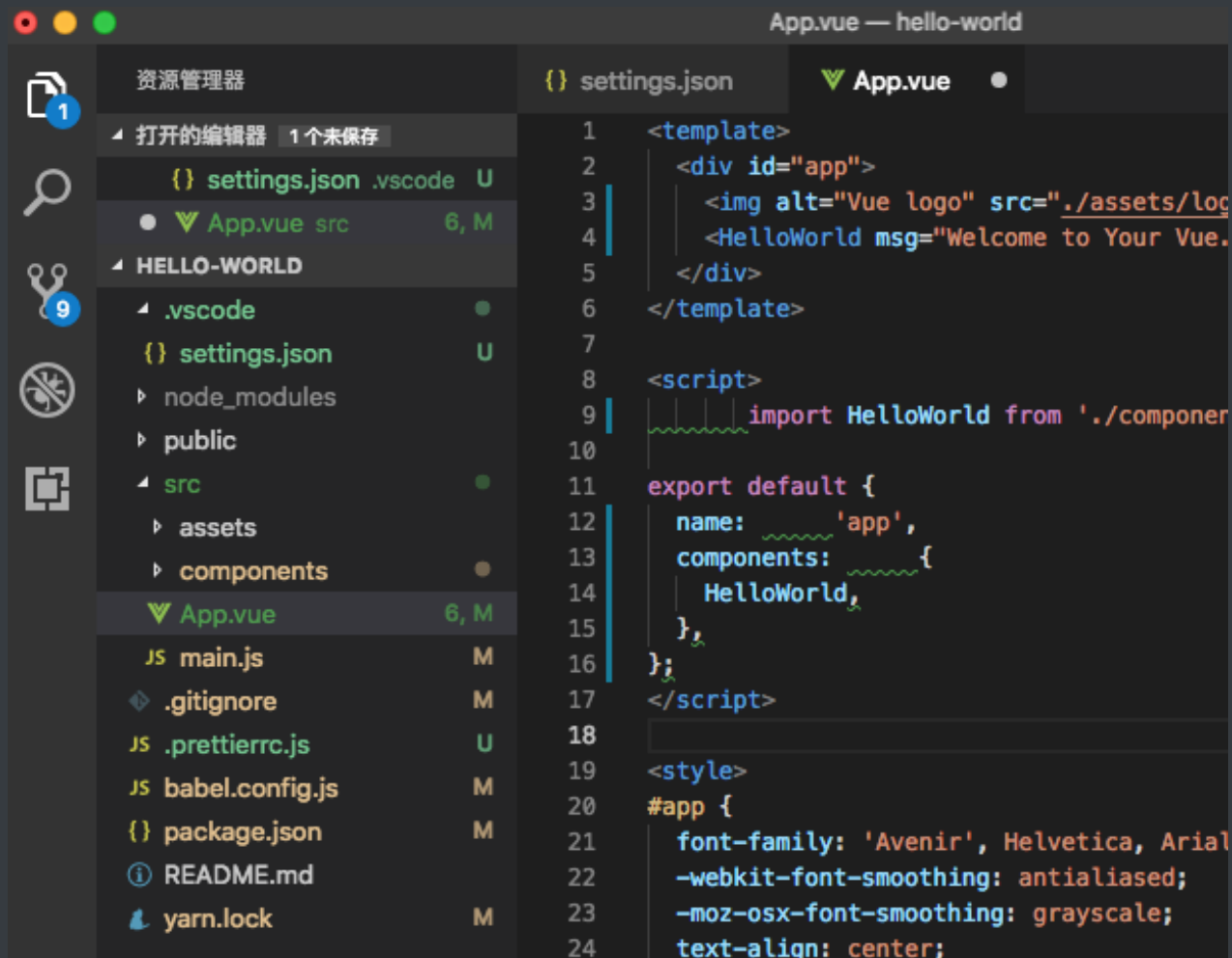
你以为安装上 `eslint` 插件就行了吗？不行的。因为 `eslint` 并不知道我们的 `.vue` 文件里面包含了 `js` 语法，所以还需要打开我们的 `vscode` 设置文件。

注意：这里一定要设置到**项目的设置**里，而不要只是设置到你自己个人的设置里，否则你团队的小伙伴随便一改又乱掉了。正确的方法是在你项目的文件夹下有一个 `.vscode` 文件夹，而 `vue` 最讨厌的地方是它居然会把这个文件夹放到 `.gitignore` 里，这个错误你必须要纠正过来，也就是说从 `.gitignore` 文件里把 `.vscode` 删掉。切切。

在你项目的 `settings.json` 里文件里添加以下代码：

```
{
  "eslint.autoFixOnSave": true,
  "eslint.validate": [
    "javascript",
    "javascriptreact",
    {
      "language": "vue",
      "autoFix": true
    }
  ],
}
```

这时候所有错误都被标注出来了，注意看左侧，一定要让这个 `settings.json` 文件是绿色的，而不能是灰色的，如果是灰色的，请检查你的 `.gitignore` 文件：



因为我们在 `settings.json` 文件里设置了 `autoFixOnSave`，所以不管多么乱的格式，只要一按 `Ctrl+S` 保存，自动就帮我们吧代码格式整理好了，是不是很方便呢？

和Prettier的冲突

有些时候我们的 `vscode` 里插件装的比较多，譬如还安装了 [prettier](#) 插件，因为我们不只开发 `vue` 项目，可能还有其它类型的 `js` 项目特别是传统 `js` 项目，需要用到 `prettier` 进行美化，而 `prettier` 的一些功能是会和 `eslint` 相冲突的，比如说我们在全局设置了 `prettier` 的 `formatOnSave`，这个功能就会和 `eslint` 的 `autoFixOnSave` 打架，为了避免这个矛盾，我们通常还会在本项目的 `settings.json` 文件里再多加几个选项，类似于这样：

```
"editor.tabSize": 2,  
"editor.formatOnSave": false,  
"prettier.semi": false,  
"prettier.singleQuote": true
```

有了这些设置，基本上 `prettier` 就不会和 `eslint` 打架了。

小结

以上就是用 `vscode` 开发 `vue` 程序的标配，并不像网上有些文章说的那么简单，不是只需要配一个 `eslint` 就能解决的事情，这里还用到了 `vetur`，`eslint` 和 `prettier`，把几个工具综合用好，才能真正达到我们的错误随时可见，保存自动修改，更正既往错误的目的。希望每个前端工程师写出的代码都如出一人之手，漂亮简洁干净。