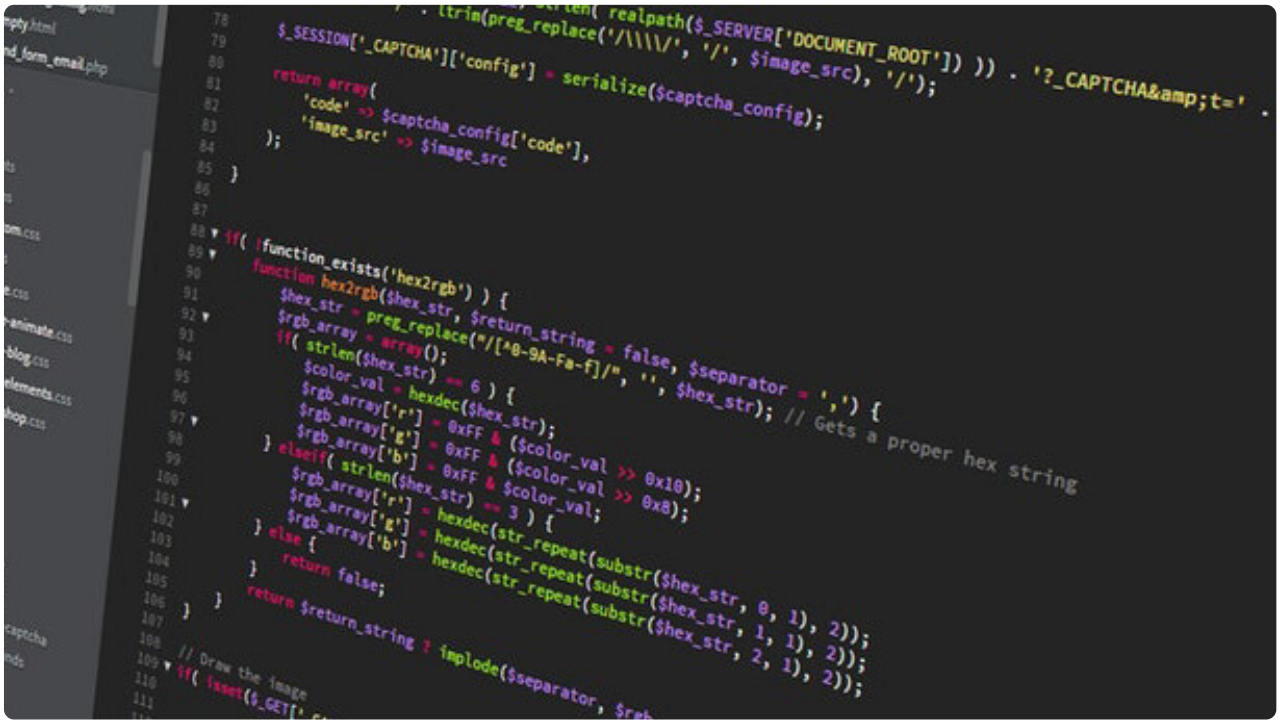
35 使用 npm 形式集成

更新时间：2019-08-16 13:44:44

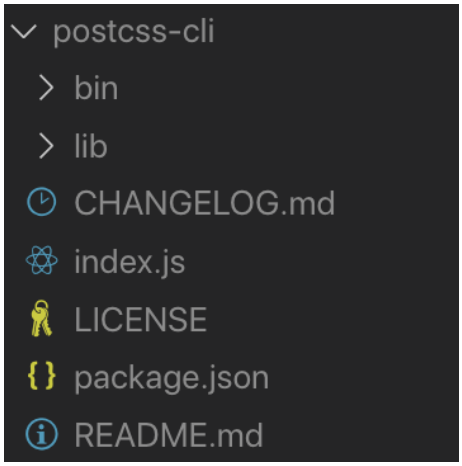


> 青年是学习智慧的时期，中年是付诸实践的时期。
>
> —— 卢梭

我们这一节用 npm 的方式来集成代码。在这一节中，我们会介绍 npm 包的结构、npm 包的制作和发布过程。

npm包的介绍

npm 是 JS 的包管理工具，我们可以用它来下载已有的代码，可以很快的引用别人发布过的代码。上一节中我们已经使用 npm 安装了几个工具，那几个工具其实就是几个 npm 包。安装好的 npm 包会被放在项目里的 node_modules 里，我们找一个包来看下它是什么样的结构，以上一节用的 postcss-cli 为例：



这个包里的内容并不多，这里面最重要的就是“package.json”这个文件，有了这个文件，就可以说这是一个 npm 包的结构了，而另外的文件就是插件功能逻辑的源码和一些说明文件。

当我们引用这个插件的时候，Node.js 就会读取 package.json 里的内容，根据 main 字段指定的路径去引用对应的文件。postcss-cli 这个插件的入口文件就是“index.js”，在“index.js”里又会引用lib目录里的代码。

这个插件还需要对外提供可执行文件，就是我们使用 npx 工具调用的 postcss 命令，所以会在 bin 目录里放上可执行文件，然后再在配置文件里的“bin”字段配置上可执行文件的目录位置。这个目录里其他几个文件，就是一些版权声明或者插件介绍这些说明性质的文件了，对代码的运行没有实际作用。

package.json 文件是一个 JSON 格式的配置文件，所有插件相关的信息都会记录在这个文件里。下来我们来介绍下 package.json 里几个比较重要的字段。

- **name**：这个字段指定 npm 包的名称，我们发布 npm包以后，包的名称就是这个字段指定的，和目录名称没什么关系。
- **version**：这是 npm 插件的版本号，每次发布的时候不能使用原来的版本。
- **description**：npm 包的描述信息，用比较简短的语言描述这个插件是做什么的。
- **keywords**：npm 包的一些关键词，用户可能会根据这些关键词模糊搜索到你的插件，如果希望你的 npm 包更容易被找到，可以好好设计一下这个关键词。
- **main**：这是npm包的主文件，通常是一个 JS 文件，但这个字段也可以是 CSS 文件。
- **license**：指定 npm 包的开源模式，是只允许调用代码，还是可以随意改动。
- **dependencies**：这是一个很重要的字段，在 npm 包中，你可以使用其他的插件，这个字段就是指定你这个包都需要哪些依赖的。只要和上一节中那样把需要的插件名称和版本填写到这个字段中，就可以直接引用了。
- **devDependencies**：这个字段和上面一个很类似，它俩的区别就是所安装插件的作用不同，dependencies 记录的通常是会被打包到最终代码里的 npm 包，而 devDependencies 字段记录的是开发过程需要用到的 npm 包。

这些就是有关 npm 包的一些介绍。

npm包的制作

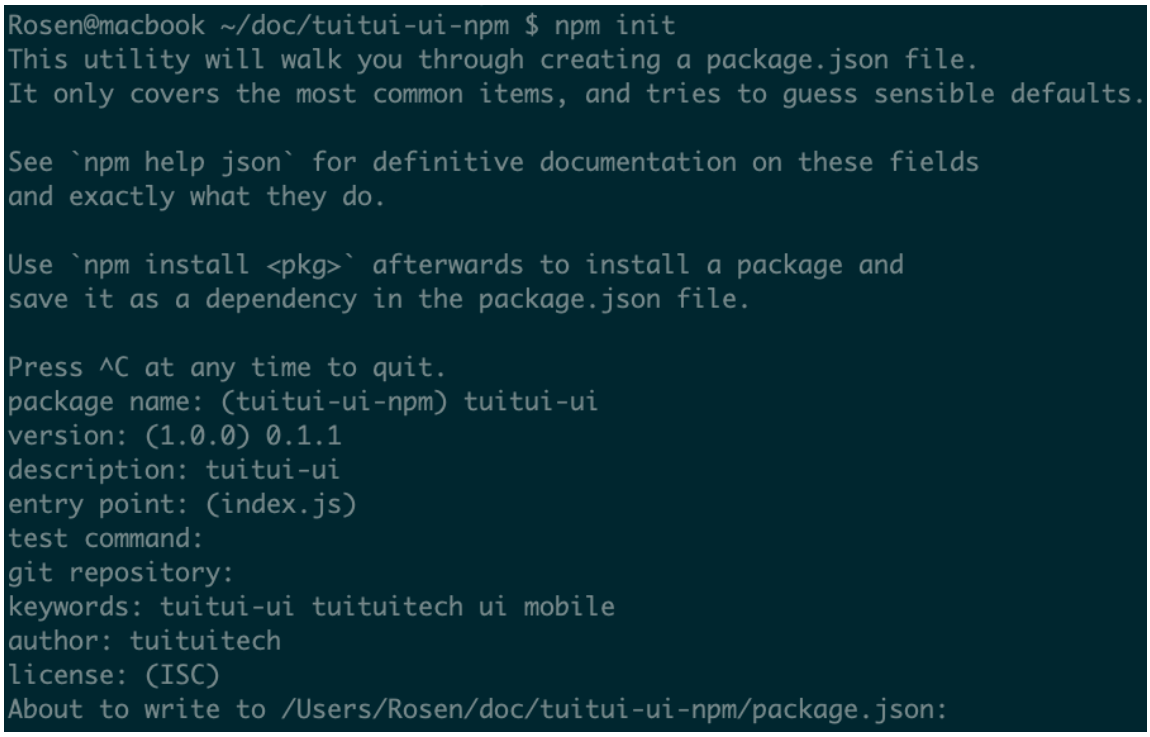
下来我们开始制作 npm 包，这个过程分三步来实现，分别是项目初始化、依赖的配置和添加文件说明。

一、项目初始化

我们这里为了减少 npm 包里的文件，不打算在原有的项目上直接进行发布，会重新建立一个新的项目。我们先建一个名为“tuitui-ui-npm”的目录，然后和上一节一样，在目录下执行下面的命令对目录进行 npm 的初始化：

```
npm init
```

执行过命令后输入关键的信息，如下图：



这样就生成好了 package.json 文件：

```
{
  "name": "tuitui-ui",
  "version": "0.1.1",
  "description": "tuitui-ui",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [
    "tuitui-ui",
    "tuituitech",
    "ui",
    "mobile"
  ],
  "author": "tuituitech",
  "license": "ISC"
}
```

现在，这就是一个 npm 项目了，这里我们把版本先定到“0.1.1”，等使用一段时间稳定后再把版本从“1.0.0”开始计算。接下来我们在 tuitui-ui-npm 目录下创建一个名为“css”的目录，然后把之前项目里生成的最终文件 tuitui-ui.min.css 移到新建好的 css 目录里。这样项目初始化的工作就完成了。

二、依赖的配置

这一步我们来配置依赖，我们之前打包好的样式是使用网络地址来加载依赖文件 font-awesome 的。但在 npm 中，要遵循 npm 的设计规则，在插件里再引用网络地址就不太合适了，所以我们这一步就是要更改项目依赖的引入方式。

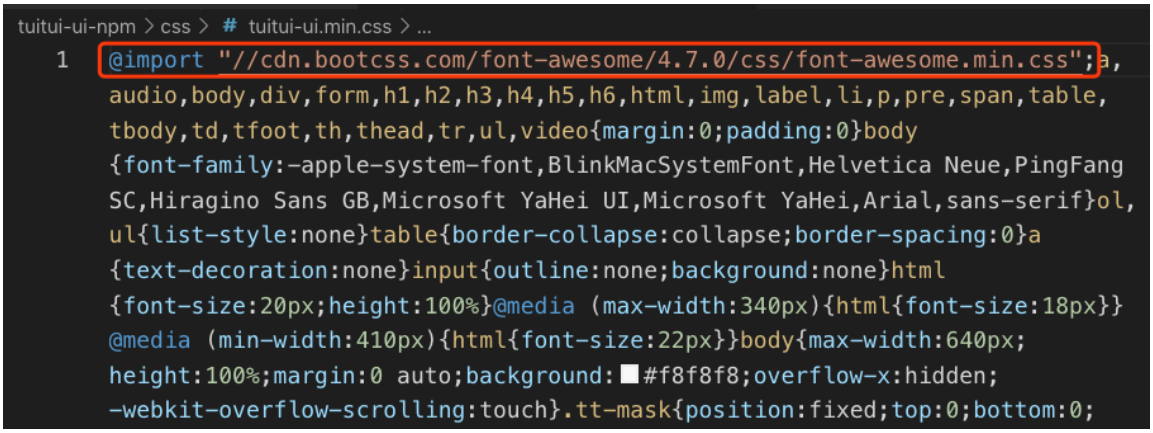
这里我们使用 npm 引入依赖的方式，首先要在新的这个项目中安装 font-awesome 依赖，命令如下；

```
npm i font-awesome@4.7.0 -S
```

-S 参数就是在 dependencies 里记录所安装插件的信息，这样项目的依赖就安装好了，package.json 里多了下面这段内容：

```
"dependencies": {
  "font-awesome": "^4.7.0"
}
```

有了这个依赖，原来打包好代码里的网络地址就可以去掉了，这时候我们可以把 /css/tuitui-ui.min.css 里的第一句话去掉。



最后一步，我们再在根目录下添加一个index.js文件，把需要的两个css文件都引入进去。在/index.js中添加：

```
/*
 * @Author: Rosen
 * @Date: 2019-08-15 22:19:05
 * @Last Modified by: Rosen
 * @Last Modified time: 2019-08-15 22:19:05
 */

import 'font-awesome/css/font-awesome.min.css';
import './css/tuitui-ui.min.css';
```

> **@ Tips：**
> 这里有两个地方要注意下：
> 1、npm工具只能通过js来引入包，不能直接用css引用其他的插件。
> 2、在index.js里引入css文件，这种引用方式只有在像webpack这种可以把CSS当做JS模块来处理的工具里才可以使用。如果是其他情况就要手动的用路径去引用CSS文件。

这样我们这个项目的依赖配置就完成了。

三、添加说明文件

最后，在这个项目里添加一个简单的说明文件，在根目录下建立一个“README.md”文件：

```
# tuitui-ui样式库

## 项目介绍

tuitui-ui，由推推科技开发的UI样式库。微信关注推推优品，回复“ui”即可获得项目的预览地址。

![](http://ui.tuituitech.com/demo/img/modal-test.jpg)

## 说明
本插件依赖font-awesome@4.7.0，安装本插件的同时也会安装font-awesome。在使用本UI样式库的时候可以直接引用对应的CSS文件：

import 'font-awesome/css/font-awesome.min.css';
import 'tuitui-ui/css/tuitui-ui.min.css';


同时，本插件已经使用js文件打包了两个css文件，在支持CSS模块的项目里也可以像下面这样直接引用插件：

import 'tuitui-ui';
```

这样我们的 npm 包就建好了，下一步我们要把它发布到npm上。

npm包的发布

下来我们做 npm 的发布，这一步其实比较简单。但因为不太常用，所以同学们可能都不太熟悉。

一、注册npm帐号

如果想在npm上发布新的代码包，必须注册称为 npm 的用户，官方地址是https://www.npmjs.com/。进入后点Join按钮就可以进入用户注册页面了。在注册页面填上相关信息，然后创建一个新的用户就行了。

二、发布前的检查

如果要把代码包发布到官方的 npm 源上，要确保我们当前使用 npm 源也是官方的。像淘宝源这些非官方源只支持npm包的下载，但会明确说明不能支持 npm 包的发布。

下来我们检查一下本机用的到底是哪个源，在命令行中输入下面的命令：

```
npm config get registry
```

我这里之前设置过淘宝源，所以会显示：

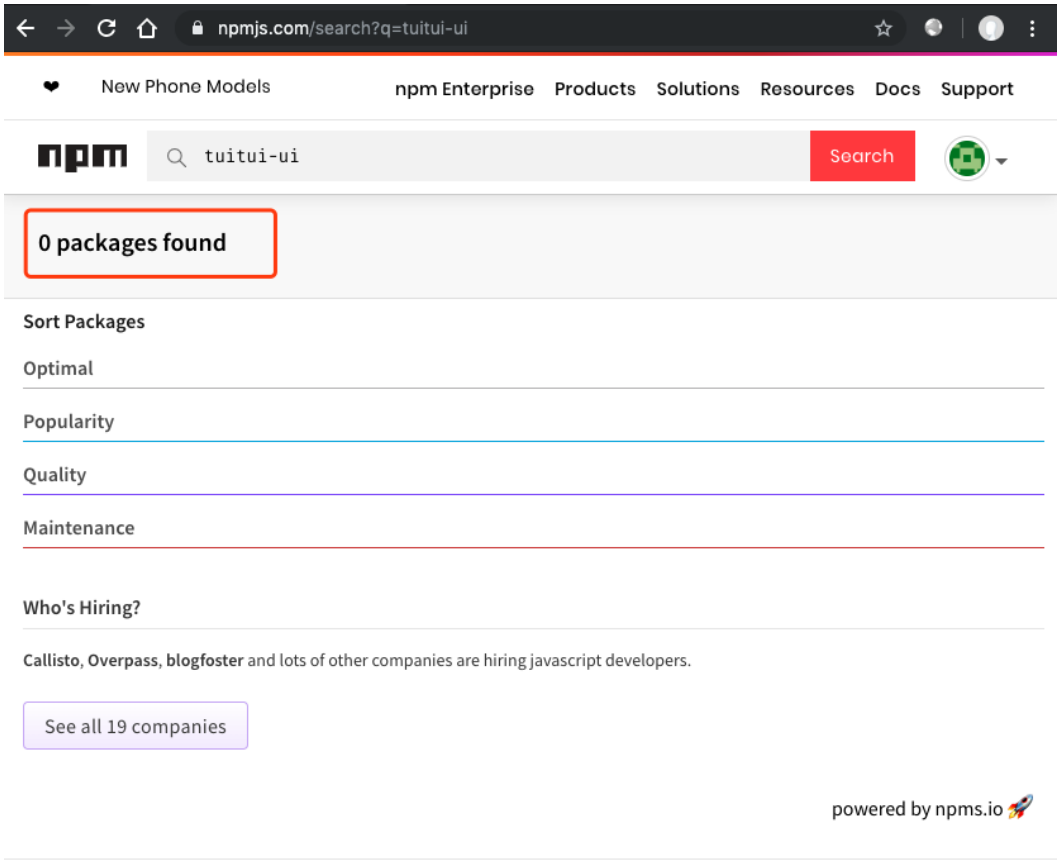
接下来我们要把它改回到官方的源地址，可以直接用命令来切换：

```
npm config set registry https://registry.npmjs.org
```

再执行刚才查看 npm 源的命令，就可以看到源被切换到官方的地址上了：

```
Rosen@macbook ~/doc/tuitui-ui-npm $ npm config get registry
https://registry.npmjs.org/
```

这样 npm 源就切换成功了。接下来还要检查下我们要发布的npm包的名称是不是有重复的，可以在npm官网上搜索一下你要发布的包名，如果像下面这样找不到重名的包，就可以进入发布工作了。



三、npm包的发布

最激动人心的一步来了，我们要把这个源发布到 npm 上了。首先要在项目目录里执行下面命令来初始化 npm 用户信息：

```
npm adduser
```

执行完命令后输入你注册时使用的用户名和密码，另外还要输入一个用于接收消息的电子邮箱地址，都完成后就完成了登录：

```
Rosen@macbook ~/doc/tuitui-ui-npm $ npm adduser
Username: tuituitech
Password:
Email: (this IS public) rosen@tuituitech.com
Logged in as tuituitech on https://registry.npmjs.org/.
```

这里要注意，只有在第一次发布的时候才需要使用“npm adduser”命令，如果以后再登录，直接使用下面的命令就可以了：

```
npm login
```

登录完成以后，就可以做最手抖的一个操作，执行下面的命令进行包的发布

```
npm publish
```

这个命令一执行成功，你的代码就发布到 npm 上了，所以这个操作要稍微慎重一点，确认好代码没有问题了再去执行。发布完成以后，会有一个很简单的提示，这看起来根本不像刚做了一个很手抖的操作。

```
Rosen@macbook ~/doc/tuitui-ui-npm $ npm publish
+ tuitui-ui@0.1.0
```

这样我们的包就发布完了，现在可以直接在其他项目中直接使用 npm下载我们的 UI 样式库了：

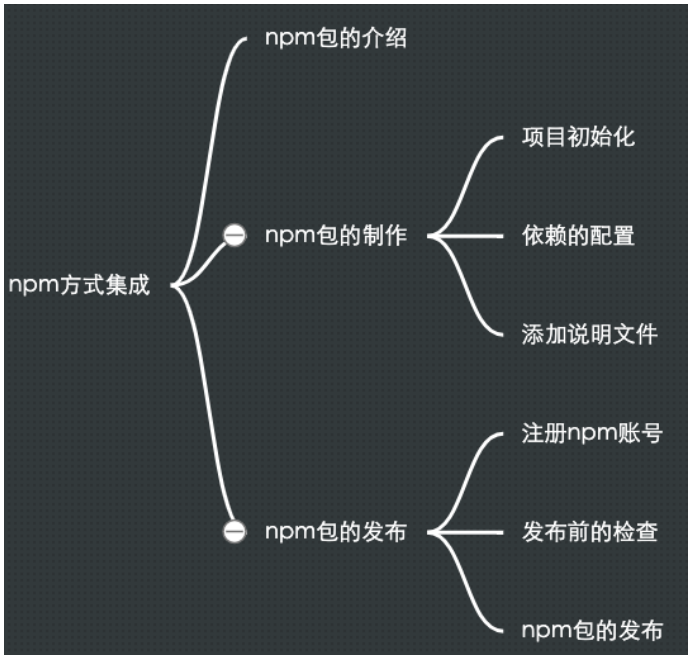
```
npm i tuitui-ui@0.1.1 -S
```

如果发现我们发布的版本有问题的话，在 24 小时以内还是可以对发布的版本进行删除。这是因为担心你发布的包被别的项目引用后，如果你删掉会导致使用你 npm 包的项目全都废掉，所以会有个时间限制。这里建议如果不是很严重的问题，最好使用“npm deprecate”命令来标记这个版本是废弃的，而不是去强行删除掉。

到这里 npm 包的发布工作就全部完成了。

总结

这一节我们做了 npm 的介绍、npm 包的制作和 npm 包的发布三部分内容。把之前使用文件形式集成的最终成果转变成了一个可以在 npm 项目中使用的插件。这一节涉及的内容又和 CSS 关系不大，如果感觉看的很迷茫，说明 npm 相关的知识还是欠缺的，这也只能下去自己多看一些这方面的知识。这一节的内容结构如下：



我们这一节的内容就到这里，这一章的操作部分也都完成了。下一节我们将会对这章的内容做一个总结。

}