

33 UI 框架的集成

更新时间：2019-08-12 11:04:43



“天才就是长期劳动的结果。”

——牛顿”

之前经过磨剑、列阵和操练的过程，整个项目核心的内容已经完成了。这一节我们开始“集结”，把之前完成的代码进行优化、整合和包装，达到可以在实战项目中直接使用状态。在这一章的内容里，会涉及到一些基于 `nodejs` 的工具，并且会使用 `shell` 脚本来完成一些辅助的工作。用到的这些内容都比较简单，如果同学们有兴趣下去可以再深入的学习一下这块的内容。

我们这一章主要包含两部分内容。第一部分是把我们之前零散的文件打包成一个最终压缩好的文件，提供给传统的项目使用。第二部分是再把压缩好的目标文件依照 `npm` 包的规则进行修改，并发布到 `npm` 源上，提供给一些基于 `npm` 的项目使用。下面来简单说下这两部分内容里都要做哪些事。

一、文件方式的集成

在以文件方式集成中，我们会处理如下几个问题：

1. 源文件中都是使用 `@import` 做的文件引入，我们需要借助工具对 `@import` 做解析，把零散的文件集合到一个 `CSS` 文件中。
2. 之前在书写样式的时候，没有写兼容的语句，这里我们要借助工具来解决兼容性问题。
3. 最后要把文件做最小化的压缩，生成 `.min.css` 的文件提供给别的项目引用。
4. 使用 `shell` 脚本完成整个集成的过程，减少命令的输入。

二、制作并发布 `npm` 包

在使用 `npm` 方式集成的时候，我们会利用上一步打包生成的样式文件。但在前面生成文件的基础上，我们会更改这个项目对字体图标库的依赖方式，以符合 `npm` 的规范。这一节我们在适配完 `npm` 包，还会把做好的 `npm` 包发布到 `npm` 官网上。这块会介绍整个发布过程，让同学们了解 `npm` 包的发布方式，以后就可以做属于自己的 `npm` 包了。做完 `npm` 发布以后，在其他项目里有用到这个框架，就可以直接从 `npm` 上下载了。

结语

这一章的内容就这两部分，会引入一些非样式相关的内容，同学们尽量去理解，有 `nodeJs` 基础的话肯定是没有问题。我们这一节的内容就介绍到这，接下来直接进入实际操作吧。

}