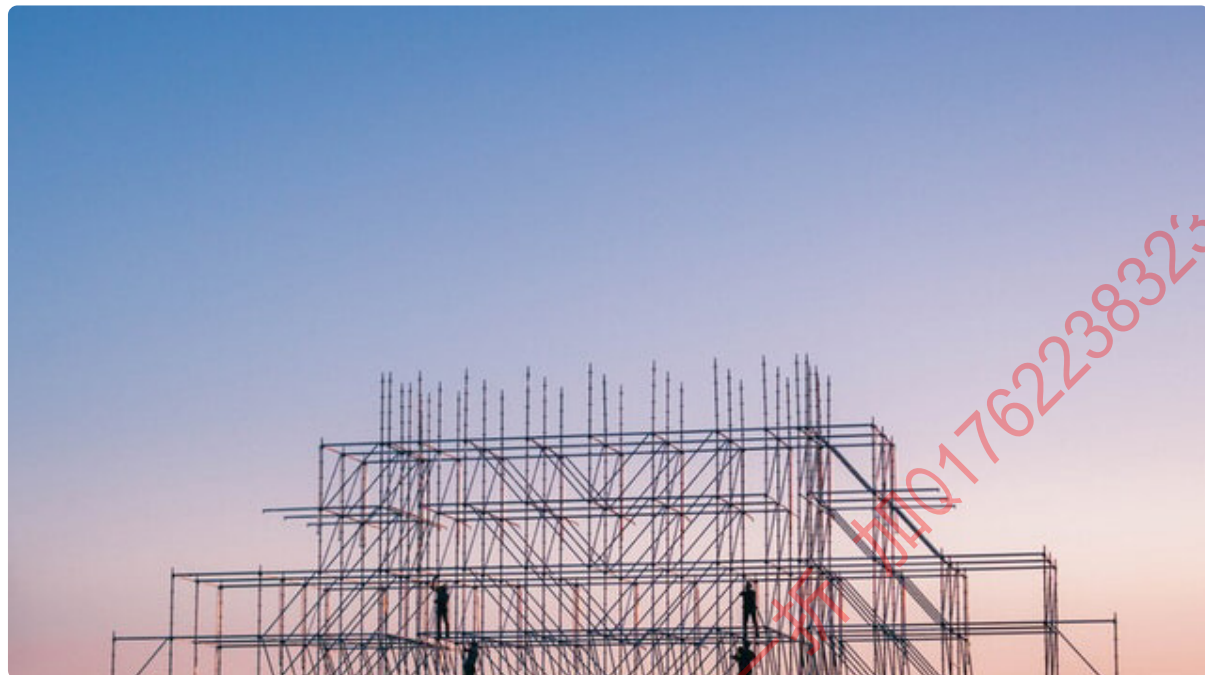


03 自己开发 UI 框架

更新时间：2019-06-21 14:22:32



“与有肝胆人共事，从无字句处读书。”

——周恩来”

上一节讲了 UI 和 UI 框架相关的知识，这一节我们来介绍下在什么情况下我们需要自己开发一套框架，而不是用已经有的那些。再就是聊一下当我们自己开发 UI 框架的时候，该怎么对它做设计。

为什么自建 UI 框架？

市面上开源的 UI 框架有很多，但一般规模比较大的公司都还会选择自建一套 UI 框架。公司愿意花费这个成本打造自己的 UI 框架，肯定是现有的开源框架不能完全满足业务上的需求。下面来说下自建框架的好处：

- **源码可以改动。**我们在使用外部框架的时候，考虑到框架要升级，所以一般要求是不改动其源码。尤其是以 npm 方式安装的框架或插件，源码更是不能动的。所以如果是自建框架的话，我们是能控制框架内部代码的。
- **可以完全满足产品的设计风格。**外部框架的样式是已经确定的，和实际产品的设计上肯定有差异。这个时候如果用外部框架就要在它上面以打补丁的方式进行样式覆盖，这种补丁多了以后也会成为一种灾难。
- **功能定制化。**每个业务都有自己注重的地方，比如去哪儿网这类网站在日历和筛选这些功能上要求比较多，滴滴可能对地图方面的要求比较多。自建 UI 框架就是可以做到定制化，让框架的设计聚焦在自己业务的侧重点上，把要求高的组件做的更精细，把没用到组件剔除掉。
- **利于风险控制。**外部框架的 Bug 也是一个比较大的问题，使用的时候遇到了只能等框架升级，或者也是用比较丑的补丁打上去。

虽然自建框架好处很多，但外部框架也是有用的。在一些对设计风格要求不高，功能相对简单的系统上，就完全可以使用外部的框架。比如我们常见的后台管理系统，它一般就是增删改查的功能，公司内部人员使用，对样式要求不高，这时候就可以考虑用上 bootstrap 这类框架进行快速搭建。

考虑使用外部框架还是自建框架，其实就是一个成本的权衡。自建框架带来的收益（要考虑到未来的收益）能覆盖住开发自建框架成本的话，那就值得去开发一套自己的框架的。

但我为什么强烈要求同学要掌握自己制作 UI 框架的技能呢？那是因为它是初级前端开发向中级前端开发前进的必学技能，想想看，你是一个大公司的前端 UI 库的开发的参与者或者甚至是发起人、主导者，你下回跳槽的薪金还怕要不上嘛。

UI框架的设计要求

当我们准备自建一套 UI 框架的时候，就要 UI 的几个贡献者一起合作，完成这个工作。在设计的时候，应该满足以下的要求：

- **友好**。之前说过，UI 最重要的作用就是让用户用的舒服。所以在开发 UI 时首先要考虑的就是让里面的组件更人性化，比如不重要且过长的文本要折叠，列表滚动要有惯性，操作有反馈但不会过分打扰，等等。“友好”主要是对产品和 UI 设计师提出的要求，但前端开发人员不能觉得这不管我的事，团队里每个人都应该有自己的想法和提建议的义务。
- **美观**。这是构成用户体验另外一个比较重要的因素。在广大的用户群体中，颜控还是不少的。美观的界面一方面能让使用者心情愉悦，同时还能体现出一个产品的精细度，提高用户对产品的信赖。
- **效率**。我们开发一个 UI 框架，最主要的动力是可以提高开发效率。这一条才是对我们前端开发人员提出的要求。我们在对 UI 框架做代码设计的时候，考虑的一定是让代码更容易复用，并且对业务范围内的情况考虑的尽量全面。这样才能让使用这套 UI 框架的开发人员节省出在 UI 上的工作量，争取早日值回票价。
- **易扩展**。最后还有一个要求，就是我们的框架要易扩展。我们在设计的时候，不能考虑到所有的业务情况，所以框架中的代码一定是有作用域限制的，容易被覆盖的，以及容易和其他框架或组件兼容的。这就要我们在写代码的时候要有一定的规范，详细的规范我们下章再聊。

小结

这一章的内容比较少，主要就讲了两方面的内容。一个是弄清楚在什么情况下我们最好自己开发 UI 框架，又在什么情况下可以使用外部框架。另外一个是要知道在设计一个 UI 框架的时候要从哪几方面去考虑，不要偏离了自建框架时的初心。如果非要给自建框架找两个关键词，那我觉得应该是“效率”和“风险控制”。