你问我答

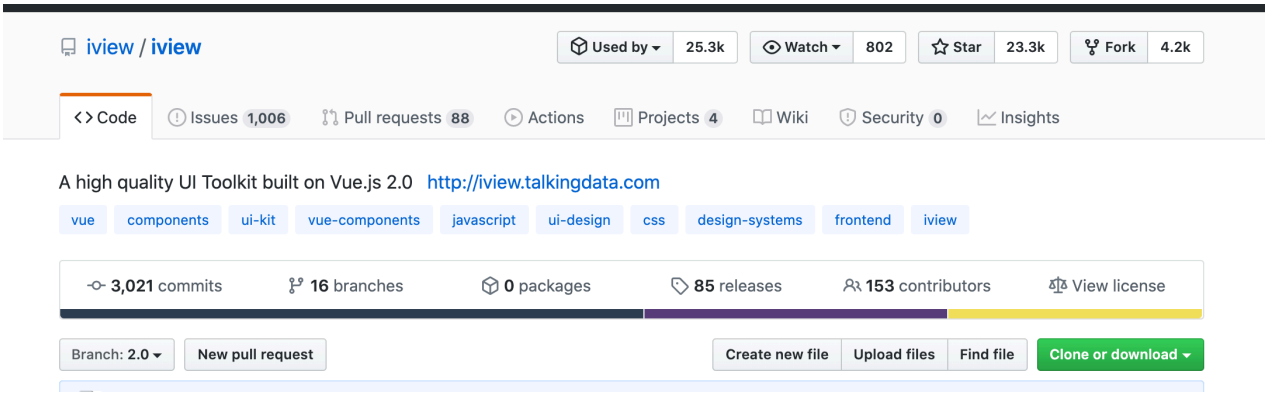
iviewUI和elementUI你会选择哪个，为什么?

iview在2019年底改了名，叫view

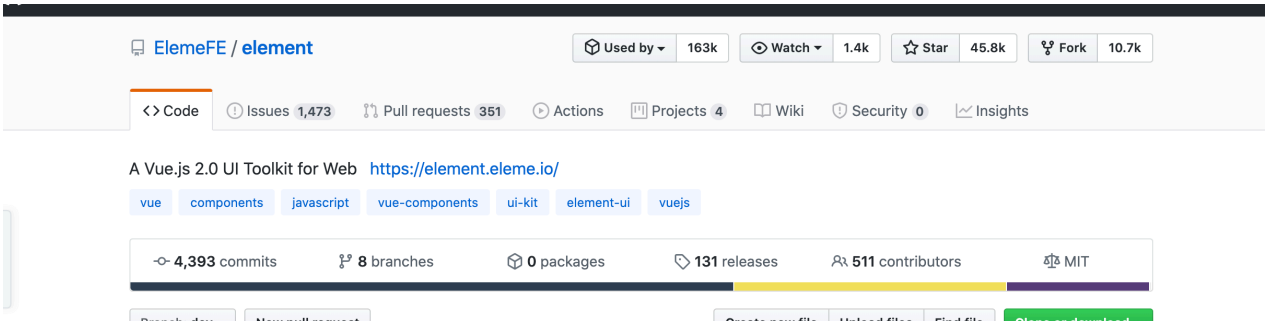
会选择iview，因为做项目初期，我也是看了很对基于vue的UI框架，因为要使用这种框架类的东西，首先还是要看里面的内容是不是满足我这个项目的需要，其次是看使用起来是否便捷。

选择这iview是因为我当时是要做一个以图片展示为主，辅助会有很多作品上传、日期选择、分类展示等等吧，我起初都有所尝试

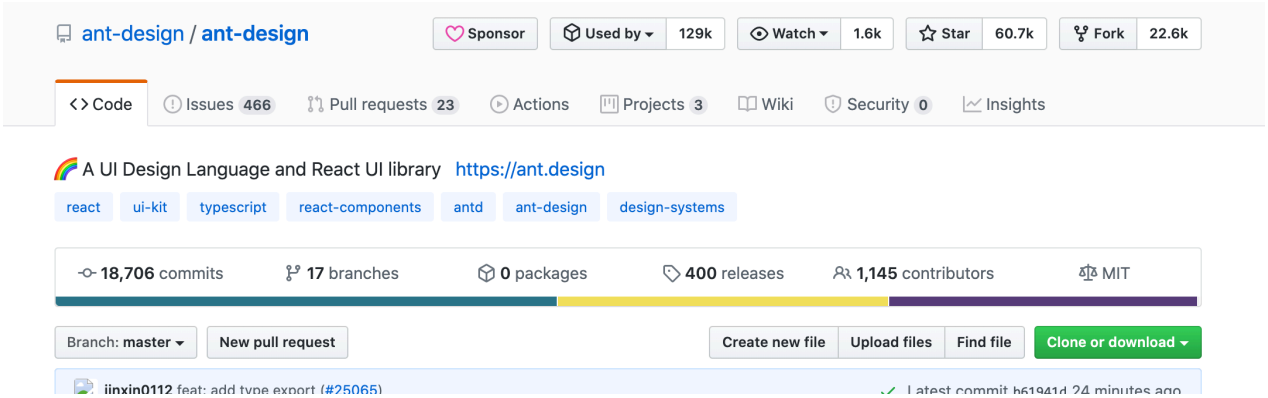
iview的文档在我看来，算是同类UI框架里最仔细的了，而且还连挂着很多讲解视频以及如何编写组件库的视频。



饿了么



蚂蚁金服



以上对比 其实可以看出， element 开发者团队规模大于iview 团队，其结果就是 无论是提交代码频率，发布版本数量 都比iview 更强！

iview的api要比element的api简洁很多。饿了么更侧重在**template**里直接去渲染模版

过渡动画

- element 有内置过渡动画 使得组件的切换变化 更具动感
- iview 更为中规中矩

对设计人员

- element 提供了 Sketch 和 Axure 工具 对设计人员友好
- iview 没有提供

gulp与grunt

更准确的讲，Grunt和Gulp属于任务流工具 `Tast Runner` ，而 webpack属于模块打包工具 `Bundler` 。

Gulp吸取了Grunt的优点，拥有更简便的写法，通过流（ `Stream` ）的概念来简化多任务之间的配置和输出，让任务更加简洁和容易上手。

gulp与webpack

常有人拿gulp与webpack来比较，知道这两个构建工具功能上有重叠的地方，可单用，也可一起用，但本质的区别就没有那么清晰。

gulp

gulp强调的是前端开发的工作流程，我们可以通过配置一系列的task，定义task处理的事务（例如文件压缩合并、雪碧图、启动server、版本控制等），然后定义执行顺序，来让gulp执行这些task，从而构建项目的整个前端开发流程。

PS：简单说就一个Task Runner

webpack

webpack是一个前端模块化方案，更侧重模块打包，我们可以把开发中的所有资源（图片、js文件、css文件等）都看成模块，通过loader（加载器）和plugins（插件）对资源进行处理，打包成符合生产环境部署的前端资源。

两者区别

虽然都是前端自动化构建工具，但看他们的定位就知道不是对等的。

gulp严格上讲，模块化不是他强调的东西，他旨在规范前端开发流程。

webpack更是明显强调模块化开发，而那些文件压缩合并、预处理等功能，不过是他附带的功能。

总结

gulp应该与grunt比较，而webpack应该与browserify（网上太多资料就这么说，这么说是没有错，不过单单这样一句话并不能让人清晰明了）。

gulp与webpack上是互补的，还是可替换的，取决于你项目的需求。如果只是个vue或react的单页应用，webpack也就够用；如果webpack某些功能使用起来麻烦甚至没有（雪碧图就没有），那就可以结合gulp一起用。

针对同一代码不断精简这个问题做一下阐述

就拿我们公司目前的一块儿广告业务举例吧，这块业务是没有公司系统可以生成使用的，但是当我做了一段时间以后，发现代码的服用性特别高，而且很多数据内容需要编辑来手动填充的，所以在复用性的基础上我也是在不断整合。

1. 从刚开始车型数据到最后图片数据车型数据还有经销商数据的归类，通过一个JSON文件完成所有的数据填充；
2. 由于数据内容繁多，最后自己又写了一个excel格式转换的网页，帮助实现自动化生成所需要的JSON文件；以上是为同事工作着想的。
3. 代码的复用性方面也是在不断整合，进行方法组件的优化，让代码看上去更整洁，方法使用起来也更方便，还有代码语义和规范方面都在逐步的养成自己的编写习惯，最简单的，就是命名方面，
    - 私有的属性和方法，前缀前面加下划线
    - 构造函数首字母大写，驼峰命名
    - 普通函数：小驼峰式命名法，首字母不大写
    - 常量全部大写，变量驼峰命名