

个人信息

- 骆也 / 男
- 博客 : <https://luoyefe.com>
- Github : <https://github.com/luoye-fe>



联系方式

- 手机 : 18005246366
- Email : luoyefe@gmail.com
- QQ : 842891024
- Wechat : l842891024

开源项目

GoH5

可视化 H5 编辑器，类似易企秀的开源项目。

-  Star 455
-  Fork 136
- 地址: <https://github.com/luoye-fe/goh5>

AMP

便捷的 API 管理平台，最大的特色是灵活的 mock service 。

-  Star 62
-  Fork 23
- 地址: <https://github.com/gomeplusFED/AMP>

FET

基于 Electron 开发的跨端桌面应用，主要为了前端开发提供多个工具。

- 地址: <https://github.com/gomeplusFED/FET>

#技术文章

- 如何写一个 NodeJs 代理服务器
- JS 中的异步流程

#演讲和讲义

- 客服平台公司内部总结
- 北京 NodeParty 8.21 关于 AMP 的分享

#工作经历

国美 (2016.03 - 至今)

进入国美后，一方面以组长的身份带领 6 人小组承接中后台类项目的开发，涉及的开发任务不仅包括前端，也包括后台服务；另一方面，与四人组成的架构小组负责推进整个前端部门的架构演进，负责开发流程，部署流程，技术推广，人员成长等工作。

具体的开发项目如下：

后台类

在国美参与了大量的内部中后台项目的开发，最后落地形式都为纯单页应用，技术栈，框架主要为 `Vue` 相关，构建主要为 `Webpack` 相关，后台服务有 `NodeJs`，也有 `Java` 等，详细情况如下：

- 数据采集平台与客户端 SDK

关键字：无痕埋点、`iframe`、选择器

运营人员为了能够最大程度定制信息采集和实现自由的埋点管理，并且不依赖开发人员手动增加埋点信息，诞生了这个无痕埋点的后台项目以及对应的客户端 SDK。

主要功能就是通过 `iframe` 获取前端页面，右击元素，以 `css` 选择器作为唯一 key 添加埋点信息，保存后可便捷的对已埋点位置进行管理。客户端挂载对应的 SDK 后，此 SDK 会向后台请求本页面埋点信息，通过选择器绑定事件，实现数据上报。

这个平台的独立性其实并不高，在开发时间紧张，开发人员紧张的情况下，我将这个平台集成到已存在的数据平台（如下述），利用现成的权限系统，路由系统，以及利用已经过实践考验的打包构建部署流程，快速搭建了这个后台系统。

信息采集平台的主要技术点包括两个方面，一是后台中 `iframe` 页面的获取，由于后台系统和公司前台并不共通，要想完全控制 `iframe` 必须通过后台服务将这个页面拿到再返回给系统前台，并且还要考虑一些 `ajax` 请求跨域的问题，涉及的细节较多，需一一处理；第二个就是 SDK 的问题，

涉及兼容性，性能问题，错误处理问题等。

- 数据开放平台

关键字：报表、自定义、可视化SQL

这个平台的目的是为运营人员提供一个自定义报表的工具，能够极大解放开发人员的效率，减少重复开发。

完成整个项目的架构规划后，最重要的核心技术就是可视化 SQL 和 报表组件的开发，可视化 SQL 负责让用户以简洁的方式完成复杂的查库操作，形成接口供报表调用，组件则是以最细粒度的方式供用户自由组合，联通数据源后，完成整张报表的制作。

最困难的地方还是可视化 SQL，只有不断的打磨才能让用户更便捷的完成数据源的定义。

- 客服平台

关键字：IM

一个供客服使用的 web im 平台，同样以 `Vue` 为基础的单页应用。

最核心技术点是 IM 部分 websocket 的逻辑，多种消息的收发，复杂网络情况的处理，本地存储的管理等，各种细节需多多考虑。

- 数据平台

关键字：数据可视化、模块化、面向配置编程

公司内数据展示平台，展示复杂多样的数据来源。

此项目属于老项目改造，之前为 express + ejs 模板形式，每次来新需求都需要前后端人员共同参与，做了大量的重复性劳动，并且开发时间较长。

改造后，前后端分离，前端的 view 层不再依赖后端，后端能够专注于性能的提升，服务的稳定等。

并且，前端部分每一页由不同模块组成，模块又由组件构成，而想要得到一个页面，只需在接口中提供此页面需要的模块和组件信息即可，模块和组件组合完成后，会自动查询数据，完成页面渲染。

这样的话，前端人员的工作就转移到了组件复杂度的提升上，而对后端来说，想要新增一个页面，只需要完成数据查询后，按照规则给前端提供一套配置信息即可，实现了面向配置编程，掩盖逻辑上的复杂度。

在这样的基础上，又提供了一套自动化工具，进一步降低开发复杂度。

举个例子，以往开发一个新的数据展示页需要好几天甚至一周，现在只需要半天，甚至几个小时。

- 其他

全职参与上述后台类项目之外，也为公司内其他部门的中后台项目提供了技术支持，如 **企业办公** 的客户端技术选型，**CMS** 的复杂表单解决方案等。

用户类

- 广告投放平台

关键字：快速、复杂表单

面向用户的广告投放系统，并且开发时间及其紧张，只有两周，在这样的情况下，主动承担起整个项目的架构工作，采用经过多个系统考验的 **Vue** 解决方案，迅速搭建好开发环境，规划好项目模块，分配开发任务，在极短时间内完成核心模块的开发。

完成核心模块的开发后，在组内其他成员继续开发周边模块的过程中，优化项目细节，完成部署流程的联通，在预期时间内完成了整个项目的开发。

技术细节不多，最多的还是表单处理，最重要的还是应征了 **Vue** 方便快捷的特点，和自身对项目把控的能力。

架构类

- 项目组开发流程

不断的优化项目组的开发流程，从最初的刀耕火种，到有模块加载器，有本地打包构建，有本地开发服务，有环境区分，有自动化部署，有性能监控，有错误报警等等，在架构组内成员的不断努力之下，项目组的整个技术在不断演进，项目的开发也逐步趋向稳定。

- 中后台类项目的快速搭建

经过多个单页应用的实践，对中后台项目的实现有了很深的积累，在此基础上完成了一些自动化工具和功能性组件，便于快速搭建类似项目。

同时，对于单页应用也有了很深的理解，优缺点，取舍之地等，对以后的工作有很深的感悟。

其他

- **Git** 的推广使用

推进前端团队版本管理从 **svn** 向 **git** 演化，包括团队内 **git** 使用的培训，相对应配套工具的提供，自动化工具从 **svn** 到 **git** 的迁移，**git work flow** 的实现等，还有推进运维部门部署 **git** 服务器，培养团队内 **git** 氛围等。

- **Vue** 的推广使用

推进团队内从 **jq/zepto** 技术栈向 **vue** 演化，当然，在合适的场景选择合适的工具才是最核心的，**vue** 作为一个渐进式框架能够适合多种场景，并且上手难度低，功能强大，无疑很受团队欢迎。

新浪 (2015.10 - 2016.3)

新浪的工作经历主要让我对大公司的流程有了清晰的认识，并且引领我真正走入前端的世界。主要负责的项目如下：

- 蒲公英后台

一个快速搭建 H5 页面的内部工具，技术栈 Angular。在已存在产品的基础上做了很多优化工作，包括新增本地构建流程，重写老的组件（尽可能遵循 Angular 的推荐模式），梳理架构优化数据结构等。

- 蒲公英前台

主要是后台的数据完成后前台的渲染工作，技术栈 Swiper 滑动库。为用户提供更优秀的体验，更多样的操作方式，更复杂的表单系统等。

美食杰 (2015.3 - 2015.10)

进入前端职业的第一家公司，主要是基础的积累。

#技能清单

以下均为我能够熟练使用的技能：

- Web 开发：Node
- Web 框架：express/koa
- 前端框架和库：Bootstrap/Vue/Electron/jQuery/Zepto ...
- 前端工具：Gulp/WebPack/Rollup ...
- 数据库：MongoDB
- 版本管理：Git/Svn