

卢晓锋

个人信息

性别：男
手机：15279189307
专业：华东交通大学通信工程

工作经验：6 年
邮箱：lxfljw@gmail.com
岗位：高级前端开发工程师

工作经历

• 深圳市兔展智能科技有限公司	2020.07~至今	高级前端开发
• 深圳市秦丝科技有限公司	2019.05~2020.06	中级前端开发
• 深圳市猜猜城科技有限公司	2018.08~2019.05	前端开发

技能

- 熟练掌握前端开发技术（HTML5、JavaScript、ES6、CSS3 等）。
- 熟练掌握 Vue、React 及相关生态技术，了解实现原理。
- 具有多端开发经验，熟练使用 Taro、Uni-app 等框架。
- 对前端工程化有实践经验，熟练使用 Webpack、Vite 等构建工具。
- 掌握计算机网络原理，对前端网络安全有一定的研究。
- 对浏览器渲染原理、页面性能优化、用户体验有了解和实践。
- 了解 Node.js、MySQL、Nginx、Docker 等服务端知识。

项目经历

1、前端业务组件库

时间：2023.06 ~ 2023.08

项目背景：前端多项目开发经常出现重复造轮子和项目代码交叉复制的问题，开发效率低下且项目臃肿。

项目描述：组件库收录公共业务组件，发布为 npm 包并提供说明文档。

项目职责：

- 技术选型和项目搭建
- 制定业务组件库纳入标准
- 业务组件库开发和代码 review

技术要点：

- 技术栈 Pnpm + Typescript + Storybook + React + Rollup，一键部署组件和文档
- 使用 @testing-library 完成单元测试
- 接入 ESLint + Lint-Staged + Prettier + StyleLint 约束代码风格和语法错误校验
- 接入 K8s 完成自动部署

项目成果：

- 项目落地实施后，累计节约了相关业务约 20% 的时间成本。
- 可以生成对外的使用文档，查找组件和成文档本降低 50%
- 以 80% 的赞同率成为团队标准文档方案

2、零售系统升级和重构

时间：2023.06 - 2023.08

项目背景：系统90%以上的项目基于Vue1开发，且Webpack1.x打包性能差，生态不完善，本地 HMR 需要约 8秒，严重影响开发效率。

项目描述：升级整个系统前端技术栈，包括 Web 端和小程序端，接入 TypeScript、ElementPlus、TailwindCSS，并重构核心功能装修页面模块，提升开发效率。

项目职责：

- 主导全部项目升级到Vue3，完成Vue Router和权限路由的兼容。
- 升级 Webpack5 和接入 Pnpm，兼容 Vite 开发构建。
- 引入TypeScript、ElementPlus、TailwindCSS和代码风格校验 EsLint，提升开发体验。

技术要点：

- 进行Vue3语法修改和生态适配，升级VueLoader，引入TypeScript、ElementPlus、TailwindCSS
- 升级到Webpack5，集成thread-loader加快打包速度，同时兼容Vite，支持分析打包体积和构建速度。
- 编写脚本将js和html合并为.vue文件，进行部分语法替换，节省约80%的手动合并时间。

项目成果：

- 升级 Vue3 生态，紧跟技术潮流，提升开发效率约 20%。
- 通过策略模式重构首页装修系统，提供代码复用率，单一文件代码从 2000 行减少到 平均 330 行。
- 本地Vite构建提升了 90% 的速度，Webpack 生产环境打包速度提升 36% 左右。
- 使用适配器模式升级路由，减少约 90% 代码改动

3、前端埋点架构设计

时间：2023.06 ~ 2023.08

项目背景：前端项目多且埋点方案各不相同且有部分入侵业务逻辑，在已有埋点 SDK 基础上，需要开发一套埋点系统。

项目描述：设计基于 React 包括 Web 和小程序的自动化埋点系统，通过配置自动生成埋点，减少埋点代码的编写量，并能通过前端脚手架快速接入。

项目职责：设计和实现自动化埋点系统的架构设计、技术选型、开发、测试、文档和部署，导出曝光埋点组件给各个项目接入，维护前端脚手架模板。

技术要点： Web 端：

- 通过浏览器事件实现路由级别埋点，并根据路由匹配数据
- 拦截 createElement 和 onClick 函数，通过元素绑定的数据进行上报数据

- 导出曝光组件，包裹需要曝光的内容，通过 IntersectionObserver API 实现曝光检测，并支持曝光配置时间后上报

小程序端：

- 代理小程序的 Page 和 Component，实现生命周期自动上报
- 代理分享事件，支持一键配置所有页面分享朋友圈
- 拦截点击函数，通过点击元素绑定的数据进行上报数据
- 同样实现一套曝光组件，由于 API 与 Web 不一致，需要重新实现

项目成果：

- 统一前端埋点方案，减少埋点代码的编写量，降低维护埋点的心智负担，提升开发效率20%左右
- 实现自动化埋点，并且点击和曝光事件做到了较低的侵入性，不影响业务逻辑
- 接入脚手架模板，快速接入埋点系统

4、小程序自动化构建和上传

时间：2023.06 ~ 2023.08

项目背景：小程序的开发和调试周期长，需要手动上传和维护版本号，多人协作容易冲突覆盖，开发和测试需要自动化来解决这些问题。

项目描述：设计自动化构建和上传小程序的方案，通过设计 CLI 方式，指定配置自动上传小程序，支持同时上传多个模板，并提供预览和检查功能。

项目职责：

- 技术选型和初始化 CLI 项目和测试脚本
- 支持上传、预览、检查、版本号管理，并提供错误自动重试机制
- 开发取码平台，获取小程序构建后的预览码或体验码

项目成果：

- 小程序开发流程简化，自动化构建代替开发人员打包和发布，开发效率提升 15% 左右
- 解决了版本号手动管理和冲突的问题，协作效率提升 60% 左右
- 取码平台提升了开发与测试的协作效率和企业微信开发效率约 30%

5、前端脚手架项目

时间：2023.06 ~ 2023.08

项目背景：

- 团队有统一的技术规范，但是没有统一的脚手架
- 前期搭建需要参考文档接入监控、Lint 等功能，耗时耗力

项目职责：

- 设计脚手架的架构设计、技术选型、开发、测试、文档和部署
- 增加询问式项目流程，减少配置项，提升开发效率

- 实现脚手架的快速搭建，包括项目创建、插件管理、版本管理、一键启动项目、给项目增加插件功能

技术要点：

- 选型 Commander + Inquirer + Ora + Exec + DownLoad-git-repo 实现脚手架的快速搭建
- 使用策略模式对应每个命令，分开维护
- 使用远程模板管理，动态拉取模板和配置

项目成果：

- 一键创建前端项目，开发效率提高约 10%
 - 一键启动项目，减少新人接入成本，项目启动效率提升 50%
 - 一键管理版本号，统一前端版本规范
-

6、移动端海报编辑器

时间：2023.06 ~ 2023.08

项目描述：该项目是移动端海报编辑器，可以通过拖拉拽的方式快速创建和编辑海报的代码平台。该项目基于 PC 端编辑器打造海报功能，支持文字、图片、SVG、员工名片等组件，并支持撤销重做功能。

技术栈：React + Redux + Typescript + Webpack + Html2Canvas

项目职责：

- 参考 PC 编辑器的设计架构，搭建编辑的基本结构
- 实现核心编辑组件，让所有组件拥有基本的拖拽缩放能力
- 实现组件的撤销重做功能

项目成果：

- 移动端可自由编辑海报作为卖点，订单率提升 10%
 - 兼容 PC 端查看和部分编辑功能
-