

卢晓锋

个人信息

性别：男

经验：6 年

手机：15279189307

邮箱：lxfljw@gmail.com

专业：通信工程

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	2020.07~2024.05	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	初级前端开发

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
- 有小程序开发经验，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
- 了解浏览器渲染原理
- 了解计算机网络原理，了解前端安全
- 了解 Node.js、Redis、MySQL、Nginx、Docker 等服务端知识

项目经历

1、前端业务组件库 2023.12 - 2024.02

项目背景：为解决前端项目开发中重复造轮子和代码交叉复制的问题，开发效率低下且项目臃肿

项目描述：创建公共业务组件库，发布为 npm 包并提供详细说明文档

项目职责：技术选型和项目搭建，制定业务组件库标准，开发业务组件库并进行代码审核

技术要点：

- 技术栈 Pnpm + Typescript + Storybook + React + Rollup，实现组件和文档的一键部署
- 使用 @testing-library 进行单元测试
- 集成 ESLint + Lint-Staged + Prettier + StyleLint 约束代码风格和语法

项目成果：

- 项目实施后，相关业务时间成本节约约 20%
- 提供对外使用文档，组件查找和文档生成效率提升 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 2023.03 - 2023.05

项目背景: 系统90%以上的项目基于Vue1开发, 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: 升级系统前端技术栈, 包括 Web 端和小程序端, 集成 TypeScript、ElementPlus、TailwindCSS, 重构核心功能装修页面模块, 提高开发效率

项目职责: 主导项目升级到Vue3, 完成Vue Router和权限路由的兼容, 集成 Pnpm, 优化依赖包安装速度, 升级 Webpack5, 并兼容 Vite, 集成 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 swc 和 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, 提升开发效率约 20%
- 重构首页装修系统, 提高代码复用率, 单文件代码从 2000 行减少到平均 330 行
- 使用 Vite本地构建速度提升 90%, Webpack 生产环境打包速度提升 36%
- 使用适配器模式升级路由, 减少约 90% 路由代码和权限配置改动

3、前端埋点架构设计 2022.06 - 2022.08

项目背景: 前端项目埋点方案不统一, 部分入侵业务逻辑, 需要开发一套统一的埋点系统

项目描述: 设计基于 React 的自动化埋点系统, 通过配置自动生成埋点, 减少代码编写量

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

技术要点:

Web 端:

- 重写 history 的路由事件, 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
- 拦截点击函数, 通过点击元素绑定的数据进行上报数据
- 实现小程序通用埋点曝光组件

项目成果:

- 统一前端埋点方案, 减少埋点代码编写量, 降低维护心智负担, 提升开发效率20%
- 实现自动化埋点, 点击和曝光事件低侵入性, 不影响业务逻辑
- 脚手架模板快速集成埋点系统

4、移动端海报编辑器 2021.11 - 2022.01

项目描述: 开发移动端海报编辑器, 通过拖拉拽快速创建和编辑海报。基于 PC 端编辑器实现, 支持文字、图片、SVG、员工名片等组件, 并支持撤销重做功能

项目职责: 参考 PC 编辑器的设计架构, 搭建基本结构, 实现核心编辑组件、拖拽缩放能力和撤销重做功能

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

项目成果: 移动端可自由编辑海报作为卖点, 订单率提升 10%, 同时也作为移动端编辑器项目进行后序维护和迭代

5、前端脚手架项目 2021.06 - 2021.08

项目背景: 团队有统一技术规范, 但无统一脚手架, 前期搭建需耗时接入监控、Lint 等功能

项目职责: 设计脚手架架构、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, 实现快速集成, 包括项目创建、插件管理、版本管理、一键启动项目、增加插件功能

技术要点:

- 选型 Commander + Inquirer + Ora + Exec + DownLoad-git-repo 实现快速搭建
- 使用策略模式维护每个命令, 使用远程模板管理, 动态拉取模板和配置

项目成果:

- 快速创建前端项目, 开发效率提高 10%
- 快速启动项目, 减少新人接入成本, 项目启动效率提升 50%
- 通过命令速接入前端规范

6、小程序自动化构建和上传 2021.02 - 2021.03

项目背景: 小程序开发和调试周期较长, 需要手动上传和维护版本号, 多人协作时容易出现冲突和覆盖的问题, 需要通过自动化来提升效率

项目描述: 设计并实现自动化构建和上传小程序的方案, 通过 CLI 工具指定配置, 实现自动上传小程序, 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, 初始化 CLI 项目和测试脚本, 实现上传、预览、检查、版本号管理功能, 并提供错误自动重试机制, 提供开发取码平台, 获取小程序构建后的预览码或体验码

项目成果:

- 简化了小程序开发流程, 自动化构建替代了手动打包和发布, 开发效率提升约 15%
- 解决了版本号手动管理和冲突问题, 协作效率提升约 60%
- 取码平台提升了开发与测试的协作效率, 以及企业微信的开发效率约 30%