

基本信息

姓名	教育情况	专业	工作经验	当前状态	求职意向
付寅生	16 年本科毕业	通信工程	8 年	已离职	前端/全职

联系方式	微信	Github	技术博客
16602835945	fridolph	https://github.com/Fridolph	https://blog.fridolph.top

专业技能

1. 掌握 HTML、CSS、JavaScript、ES6 及 TypeScript 语法，熟悉 HTTP 协议；
2. 熟悉 Vue 全家桶，ElementUI、iView、NaiveUI、AntdVue；熟悉 Vue3 新特性及 Composition API；
3. 熟悉 TailwindCSS，预处理器 Sass、Less、Stylus 及 PostCSS；
4. 熟悉原生微信小程序，使用 uni-app 进行开发；
5. 熟悉 响应式设计，对 移动端 及 PC 端各尺寸下的响应式设计有比较深入的理解；
6. 熟悉 Webpack、Vite 常规配置、loader、plugin 及 性能优化 相关配置；
7. 熟悉前端单元测试，为团队制定 Vue 组件单元测试 规范及流程；
8. 做政府项目，对于 Web 安全 有比较严格和全面的实践；
9. 熟悉 Linux 常见命令及开发环境；
10. 了解 Node.js，能用 Koa、NestJS 做基础的服务端开发；
11. 了解 Gitlab CI 流程配置，并配合 Jenkins 完成前端项目的 部署 和 上线 工作；

工作经历

成都一蟹科技有限公司（2024.03 —— 2024.08）

职位：前端主管，作为前端主管，我负责：

- 项目管理：参与需求分析与规划，制定并监控项目计划与时间表，运用语雀、蓝湖和云效工具进行团队管理；
- 团队协作：领导并协调小组成员确保项目按时完成，持续跟踪进度，解决问题，维护项目质量；
- 技术领导：制定并执行项目架构升级策略，促进技术前沿的同步和团队技术标准一致性；
- 质量控制：推动实践最佳编程习惯，通过提升代码质量与系统安全性来减缓技术债累积；
- 知识共享：定期组织内部建设和知识分享会，以促进团队成长和知识传播；

主要内容：

- 团队绩效提升：规划并实施 2024 年二、三季度计划，提高了团队的整体绩效；
- 前端体系化建设：建立并丰富语雀知识库，为后续管理和共享打下坚实基础；
- 质量保障：定期组织 CodeReview 和技术分享，提高团队交流氛围；
- 业务贡献：负责并参与合规中心迭代开发，贡献 3 个业务组件；
- 其他实践：负责研究并推进悬壶小程序的 uni-app 重构方案。

项目经历

云药客 SaaS 系统

一个为药械企业提供的业务推广解决方案，支持多组织、多任务的结算管理，确保全链合规性。

技术栈：Vue3 + antdVue + Pinia + TypeScript + ECharts

主要贡献：

- **技术升级**: 成功将系统中的 Vue2 代码和组件重构为 Vue3 + TS, 并升级至最新的 antdVue 版本以解决兼容性挑战;
- **效率提升**: 通过引入 pnpm 和转换项目结构至 monorepo, 实现了分支的统一管理和依赖管理的优化;
- **代码质量**: 加强代码的可读性和维护性, 通过添加 JSDoc 注释及优化路由和权限验证逻辑;
- **组件共享**: 设计并实现可复用的业务组件如“组合式卡片”、“模式入口”、“组合搜索”, 并成功集成到组件库中。

悬壶医讯

为医生打造的互动交流平台, 旨在激励创造价值并增强用户参与度。

技术栈: 原生小程序 (采用 uni-app 进行重构)

主要工作:

- 功能迭代: 负责小程序中“真实世界 RWS”和“抽奖活动”模块的开发与优化
- 架构优化: 参与小程序 uni-app 重构方案调研及推进 (Vue3 + TS + tailwind + uView UI)

成都网思科平科技有限公司 (2017.07 —— 2024.01)

担任角色: 前端组长

工作描述与职责:

1. 负责项目**前端架构**搭建, 确定技术框架, 明确团队开发规范;
2. 整理产品需求文档, 编写**前端开发文档**;
3. 配合设计人员进行**前端开发**, 高效完成产品与客户的需求与功能;
4. 与后端工程师共同协商约定数据接口, 完成前后台**数据交互**;
5. 进行**单元测试** 和 Code Review, 及时反馈测试相关问题;
6. 作为团队 Leader, 参与**团队建设与资源共享**。

项目经历

EDR - 终端威胁侦测与响应平台

该项目阐述 EDR 概念本身, 通过探针实时监控用户终端的进程和文件, 并实时上报汇总到服务端, 最后统一在 Web 端展示给安全专员查看。实现了**恶意代码预警和预测**, 极大地提高了客户的安全级别。我作为前端负责人, 主要负责项目 Web 前端部分的开发和维护工作。

技术栈: Vue + iView + Vxe-Table + vue-i18n + ECharts + D3.js

主要贡献:

- 负责该项目**前端架构**, 包括完善了基础模块、utils, 编写了详细的文档和示例代码, 引导团队遵循最佳实践和开发流程
- 实现首页安全概览及**可配置化展示**, 提高了用户体验和客户满意度
- 完成资产详情页面, 通过 **Websocket** 获取并展示在线终端相关信息, 提高了性能和实时性
- 封装了接口请求、高级搜索、过滤器、PowerShell 等**业务组件**, 提高开发效率
- 为告警、进程等页面请求大量数据时设计了前端过滤和预缓存方案, **提升了页面加载速度和性能**
- 自定义页面及 html、pdf、excel **导出功能**
- 负责了该 ToB 项目私有**部署方案**及相关性能优化

项目难点:

- 高级搜索和筛选器 (表单项较多, 需数据联动, 还需考虑安全性, 屏蔽敏感数据) **设计时有复杂度**
- 使用 vxe-table 实现**复杂表格** (接口信息量大、展示内容较多、列数多, 用户操作密集) 并为用户定制化 table 列与其他数据的展示
- 单个告警接口数据过多时 (超过 k 条), 考虑二次处理之后的展示和渲染, 使用 **Web Worker** 并行处理提高性能

Admin - 综合管理后台

作为公司的核心项目, 综合管理后台旨在构建一个集中式的用户权限和子系统管理平台。通过该系统, 管理员可以统一管理用户

作为公司的核心项目，项目已部署至公司内部统一平台，供相关部门使用。通过该系统，管理人员可以统一管理平台权限，为不同子项目定制功能模块展示，实现对多个子系统的综合管理，并集中进行权限相关的管理操作。此外，系统还支持免登录跳转至其他子系统，极大地提升了用户体验和管理效率。

技术栈： Vue3 + TypeScript + ECharts + Naive UI + vue-i18n

主要贡献：

- 主导并完成**项目架构升级 Vue3**，包括相关配置及构建工作，提高开发效率
- 定义主题变量，为团队编写了标准的组件规范，提高团队代码的可读性和一致性
- **封装业务组件**：通用表单、上传文件和高级搜索等，提高了开发效率，并保持用户界面一致性
- 设计并开发系统权限模块，包括路由权限、菜单权限
- 设计管理员和客户领导等角色，不同角色分配不同权限，**保障系统安全性**
- **实现核心功能模块** 定制表单及其校验，实现搜索与相关列表、图表、模版等的展示与联动效果

LC 安全分析大屏

一个综合性的安全可视化项目。通过结合 EDR 项目的数据，利用大数据分析和机器学习等技术对资产、告警和终端信息进行深度处理和分析，用户可以通过大屏快速了解安全态势，发现潜在威胁，并做出及时响应，从而提高企业的整体安全防护能力。

技术栈： Vue3 + TypeScript + ECharts + D3.js

主要贡献：

- **性能优化**：为减少加载时间，减少图片使用，采用 CSS 绘制和 SVG 来呈现
- **ATT&CK 热力图实现**：同时考虑到小屏提供了 simple 模式展示
- **响应式设计改进**：针对 lg 尺寸屏幕，对非核心展示元素进行缩略化处理，既保证页面的简洁性，又保留核心信息，得到客户好评

项目难点：

- 首屏加载，白屏优化，相关性能优化
- ECharts、d3 绘制各个子模块
- 数据响应式，各图表渲染实时更新
- 布局响应式，从平板、PC 到大屏设备展示支持

团队建设

wskp-ui

wskp-ui 是一个使用 VuePress 搭建的综合性内网在线文档，它集成了 UI 规范、代码规范、demo 展示以及其他各项团队约定。该项目是利用闲余时间自发发起的，主要目的是为产品和设计同事提供一个统一、便捷的参考平台，以便更好地进行沟通和及时反馈。

核心内容：

- 团队 UI 规范制定：明确了公共变量名、主题色值等关键要素
- 前端代码规范：团队建立统一代码规范，确定代码风格一致性
- 业务组件二次封装与示例：为产品和 UI 提供在线示例，便于快速理解
- 单元测试：以基础组件和 utils 为粒度，进行单元测试
- 配套文档编写：便于新人和同事快速理解项目结构和上手开发

wskp-mock

一个基于 easy-mock 的内网部署方案，用于模拟后端 API 接口并返回预设的假数据。这一举措旨在解决开发过程中前后端进度不同步的问题，使得前端开发者可以在没有真实后端接口的情况下，依然能够进行高质量的开发工作。

wskp-deploy

一个简易的自动打包快速部署方案，方便开发和测试人员。选择好项目及分支等配置后，自动打包生成 Dist，自动分配 IP 和端口并部署到内网，便于快速演示和测试。

四川爱礼科技有限公司 (2016.01 —— 2017.07)

担任角色： web 开发与 移动端 webview 开发

工作内容与职责：

- 参与公司主项目界面开发和 爱礼官网页面开发
- 根据提供的设计图进行高精度还原，确保前端页面与设计图的一致性
- 将页面各子模块拆分为可复用的组件，提高开发效率和代码可维护性
- 针对主流移动端设备进行前端页面的响应式适配
- 与后台开发团队紧密协作，完成前端页面与后台接口的对接工作

环球礼仪知识平台 - 关于礼仪的富媒体内容平台

传统内容型社区综合网站，类似简书、小红书。**传统多页面开发模式：** jQuery + Bootstrap + 各种 jQuery 插件。16 年底为优化平台内容，进行架构升级。尤其是前端进行大重构：

技术选型： Webpack + React + react-router + Redux

主要贡献：

- 负责静态页面及相关样式的编码工作，确保页面在视觉效果和用户体验上均达到高标准
- 封装了一系列实用的工具模块，如滚动效果、全屏特效、工具栏、动画和过渡相关的 mixin 等
- 通过响应式设计和兼容性处理，实现不同设备和浏览器上的无缝切换和一致体验

主要难点：

- 由于底层框架的变更，面临了大量的代码改动、结构调整和技术适应
- 引入 JSX 和 Redux 带来了全新的开发模式和思维转变
- 早期 React 与 jQuery 的集成并不顺畅，需解决两者之间的冲突和兼容性问题

成长与收获：

- 前端工程化开发模式的转变： 从传统静态页面开发模式过渡到前端工程化开发模式
- 深入学习 Bootstrap 框架，掌握了其栅格系统、组件样式和响应式设计等核心特性
- 熟练掌握了响应式布局的原理和实现方式，为用户提供更好的视觉与交互体验
- 改变传统 DOM 操作的思维方式，理解状态改变视图的核心思想
- 改变了传统 DOM 操作的思维方式，开始理解并实践状态改变视图的核心思想

其他

个人项目

- [my-element-plus](#) Vue3 + Vite + TS4 学习 ElementPlus 组件库
- [my-program](#) 上述学习仓库的一些实际案例，主要展示了一些较为简单的 CSS、JavaScript 效果的细节实现
- [fridolph](#) 该项目整理个人关于前端相关学习资源、教程、文章、书籍等，也作为写博客的参考
- [FE-parepare-interview](#) 在线 vitepress 文档，前端综合知识体系相关整理

参与开源

- [my-resume](#) 同下面文章, Vite + Vue3 + TS + TailWindCSS 搭建
- [MDN - 感知性能](#) 翻译文章，并根据个人理解进行润色，通过 PR
- [hexo-theme-butterfly](#) 添加预请求；优化标题样式。

最近文章

- [TailWindCSS + Vite + Vue3 打造个性化在线简历项目，从开发到上线](#)
- [什么是 Accessibility - 无障碍](#)
- [【源码学习】Radash](#)

致谢

感谢您花时间阅读我的简历，我是付寅生 ^_^ 一个喜欢羽毛球、动漫和音乐的前端工程师。期待能有机会加入您的团队！再次感谢您的关注与信任！