

张云鹏

📞 152-2324-6130 · ✉ 314705487@qq.com · 🌐 blog.claude-hub.cn · 🐙 [claude-hub](https://github.com/claude-hub)

个人信息

- 男, 1997, 27岁
- 求职意向: 高级前端工程师 - 成都
- 工作经验: 5年

教育经历

- 本科, 重庆理工大学, 软件工程专业, 2015.9 ~ 2019.7

技能清单

- 熟练掌握前端开发技术 (HTML5、CSS3、JavaScript、TypeScript、ES6等)
- 深入理解 React 及相关生态, 熟悉 React 实现原理, 从 0 到 1 编写 mini-react, 实现 fiber 架构, diff 算法等
- 对前端工程化有深入理解及实践, 熟悉 Webpack、Vite, Eslint、lerna 等工程化构建工具
- 对浏览器渲染原理、性能监控、页面性能优化、用户体验及可访问性有深入了解及实践
- 掌握计算机网络原理, 对前端网络安全、前端网络监控优化有一定研究
- 深入研究微前端 Webpack Module Federation、qiankun 实现原理, 有微前端实践经验
- 熟练使用 node, 开发 cli 工具, 以及各种提效小工具 (循环依赖检查、中文漏翻)
- 了解 SSR、Electron、Flutter、uni-app

工作经历

神策网络科技有限公司, 分析云 - A/B 测试, 前端工程师, 2021.03.11 ~ 2024.09.14

- 负责 A/B 测试系统的开发、公共组件开发, 参与前端基础框架、CI/CD、工程化建设
- 围绕前端技术债、业务痛点方向, 基于前端工程化全链路对系统进行持续的技术迭代优化, 利用技术产生更大的业务价值

成都聚思力信息技术有限公司, GUI 部门, 前端工程师, 2019.03.11 ~ 2021.03.11

- 负责 To B 在线销售平台开发, 公共组件规范制定等

项目经历

神策数据

- **SEF 基础平台框架 (2021 ~ 2024)**

项目描述

在 2021 年公司平台化战略下, 神策大前端主导开发了 SEF (Sensors Extendable Frontend), 基于 Webpack 5 Module Federation 的微前端架构, 设计并实现了 SEF 的 toolchain 和 runtime 两部分, 支持模块间的依赖共享及插拔式应用服务。通过 SEF 提升开发效率, 增强系统灵活性与可维护性。

个人职责

- 主导开发设计 toolchain 接入 monorepo 功能, 并完成 cli template 生成。最初的业务单元引用多个组件模块, 一个组件模块为一个 git 仓库, 每次修改组件需要手动升级对应的 version, 一次流程加 CI/CD 跑下来, 需要至少 2 小时。接入 monorepo 后, 合并到一个仓库管理, 并且自动更新 version, 从 2 小时的繁琐流程中, 减少到 10 分钟。

- 在 0.3 版本中，通过专项优化工作，将事件分析模块的加载体积从 19MB 减少至 14MB。主要措施包括：移除不必要的 antd4 share 包；优化图标库，仅保留使用中的图标；拆分国际化语言文件，实现按需加载；统一 lodash 引用为 lodash-es，避免重复加载，从而减少体积；在 0.3 版本中不再提供 lodash，以进一步减小加载体积和时间。在加载完主应用之后，在各业务单元加载之前，通过额外加载一个 module 对 lodash 进行 provide，从而使得在过渡期平滑过渡。

• A/B Testing (2021 ~ 2024)

项目描述

利用神策数据的 A/B 测试功能，对比分析不同版本的用户体验效果。通过实施多种类型的试验（编程、多链接、可视化、多人群、时间片、父子），对关键性能指标（如转化率、点击率）的数据收集与分析，为企业提供优化产品决策的支持。

个人职责

- 负责基础框架的接入、升级及 CI/CD 等工程化建设的优化。
- 主导开发了基于 iframe 技术的可视化试验功能，使客户能够直接修改客户页面元素，并与 SDK 团队合作开发了 iframe-bridge 工具，以促进业务单元与 SDK 之间的高效通信。
- 应用懒加载、包体积优化、统一包版本、预加载等技术手段，将首页加载时间从 5 秒优化至约 1 秒。
- 确保业务代码质量，减少线上 bug 数量，核心代码单元测试覆盖率超过 95%。
- 封装了 OperationTable、虚拟列表、NiceDrawer 等组件，减少重复工作，提高代码可维护性、易用性。
- 采用图片懒加载及占位符技术解决可视化截图体积过大的问题，优化了试验配置页面的渲染速度，将其加载时间减少了 1 秒；同时引入异步截图方法，改进了试验配置保存功能，在任何网络环境下均能实现快速保存，显著提升了用户体验。
- 封装 request 库，监控后端接口性能，包含异步接口，并推动后端同学优化接口，到目前已优化不下 10 个接口。

• SensD UI 基础 UI 组件库 (2022.1 ~ 2022.5)

项目描述

基于 Antd 4.18.2 fork 开发，以解决现有业务平台设计体验不一致的问题。该项目旨在替代 Antd 组件，减少因风格差异导致的设计还原难度和补丁代码累积，从而降低维护成本。同时，通过创建自定义组件库，确保向后兼容性，便于新功能的引入和缺陷修复，提升整体开发效率和产品质量。

个人职责

- 负责开发维护：Select、Tree、Tree Select、Select Panel、Select Transfer 等公共组件。

重难点

由于神策设计风格和 Antd 有差异性，开发之初，需要对组件原有 api 的影响。维护时，需要向下兼容，避免对业务线造成影响。比如 Select 下拉菜单新增 全选、取消、确认功能，下拉中可以搜索等。

• 提效工具开发与维护

- **pig**：开发并维护了解决循环依赖与影子依赖的扫描工具。该工具帮助团队快速识别并解决复杂的依赖问题，提升了代码质量和项目的可维护性。
- **intl-check**：创建并持续改进用于检查国际化翻译遗漏的工具。通过自动化检测，确保了产品文本的一致性和完整性，减少了人工检查的工作量。
- **feedi**：接入基于 rrweb 拓展开发的异常诊断工具，用以收集和分析客户在线上使用过程中产生的数据。此工具显著提高了对客户反馈问题的响应速度和解决效率。

- **STELLR 平台**

项目描述

STELLR 是对 B 端的在线销售平台，主要为 reseller 提供线上产品的销售服务。整个系统包含: 合同管理，用户管理，消息中心，dashboard，埋点等。项目采用 react, dva, redux, immutable 等。采用分页分包技术，单独部署每个包，减小项目大小。solv-ui 基于 ant design 3.x 版本打造，扩展出符合公司业务的精美组件，solv-models 管理业务逻辑，solv-utils 提供基础方法。

个人职责

- 主要负责项目中的用户管理模块，合同管理，dashboard 等的开发与维护。
- 参与前端组件规范讨论、组件开发，收集组件场景等。

重难点

负责大文件上传的封装，采用分片上传，断点续传，秒传，超时重试等技术，解决上传相关的一系列问题。

个人评价

- **热衷钻研与分享**：对前端技术充满热情，积极钻研最新技术和最佳实践，并乐于在团队中分享知识，促进共同成长。
- **关注前沿动态**：始终保持对前端领域新技术和新兴 Web 标准的高度关注，具备强烈的求知欲和自我驱动力，致力于不断提升个人技术水平。
- **团队合作精神**：擅长与团队成员沟通协作，能够在多元化的环境中发挥积极作用，推动项目顺利进行。
- **热爱生活**：在工作之余，也是一位资深铲屎官，热爱生活，注重工作与生活的平衡。