



姜展超

一只只有追求、有梦想、喜欢写出漂亮代码的程序猿儿！



...

25岁 | 18631012069 | hounaoshao@163.com | 塔里木大学 | 本科

求职意向

前端开发工程师

北京

随时到岗

知识储备

- 1、熟练掌握 HTML5，熟悉语义化标签，如 header、main、footer、section 等
- 2、熟练掌握 CSS3，熟悉过渡、转换等 CSS3 新特性，熟悉 W3C、IE 合模型，了解 BEM 命名规范
- 3、熟练掌握 JS，熟悉 JS 原型、原型链、闭包、调用栈、任务队列等原理，熟悉 ES6+ 新特性
- 4、熟练掌握 Vue.js、Vue Router、Vuex，并能够熟练地开发 Web 单页面应用和组件
- 5、熟悉 Element UI 前端 UI 框架，能够配合 Vue 熟练地开发页面和组件
- 6、熟悉 Webpack 打包工具和 Babel 转译工具，了解常用的配置
- 7、熟悉 Echarts 图表框架，并能熟练地做出饼状图、柱状图、折线图等图表
- 8、了解 Java、Rust、Node.js 服务器端语言

工作经验

2019.4-至今

北京爱亲科技股份有限公司

前端开发工程师

参与过耘链、耘平台、审批配置、钉钉审批、门店生命周期、爱亲供应链、营运中心、结算中心系统的开发，担任耘链、耘平台、审批配置、钉钉审批系统的前端维护负责人，独立开发过审批配置项目系统

2018.10-2019.3

深圳云天励飞技术有限公司

前端开发工程师

参与公共安全系统深目系统的前端研发与维护，该系统上过蒋昌建老师的《智造将来》节目，当时协助民警破案 1 万余起，荣获亿万人流、秒级定位成就

项目经验

2019.12-2020.1	审批配置	前端独立开发
<p>该项目用来处理公司内部各个系统产生的业务审批单以及各种权限的配置。</p> <p>该系统采用 Vue + Vue Router + Element UI 技术栈实现，项目打包通过 Webpack，代码转译通过 Babel。菜单图标采用 Icon Font，根据 BEM 命名规范编写全局通用组件样式，通过 Export2Excel 插件实现导出 excel 功能，与后端配合实现菜单权限，通过编写 Vue 指令实现操作按钮权限，通过原生 HTML、CSS、JS 封装上传文件、下载文件与预览图片等组件。</p>		
2019.10-2019.11	钉钉审批	前端维护负责人
<p>该项目是用来方便员工在手机端处理公司系统产生的业务审批单，而不用必须上电脑 Web 端的系统。</p> <p>该系统采用 Vue + Vue Router + Cube UI 技术栈实现，项目打包通过 Webpack，代码转译通过 Babel。根据 BEM 命名规范编写全局通用组件样式，前端通过钉钉 API 与后端进行配合实现钉钉的消息通知。</p>		
2019.9-2019.11	耘平台	前端维护负责人
<p>该项目是给公司的供应商使用的，用来管理供应商的信息、商品信息、库存信息、采购信息以及销售等信息。</p> <p>该系统采用 Vue + Vue Router + Element UI 技术栈实现，项目打包通过 Webpack，代码转译通过 Babel。菜单图标采用 Icon Font，根据 BEM 命名规范编写全局通用组件样式，通过 Export2Excel 插件实现导出 excel 功能，与后端配合实现菜单权限，通过编写 Vue 指令实现操作按钮权限以及输入内容规则验证，通过原生 HTML、CSS、JS 封装上传文件、下载文件与预览图片等组件。</p>		
2019.6-至今	耘链	前端维护负责人
<p>该项目是整个公司的供应链系统，系统功能繁多。其中系统功能包括商品的进销存信息、SKU / SPU 管理、供应商的信息管理、质检报告管理、库存管理、采购管理、审批管理、分析报表、统计报表等等。</p> <p>该系统采用 Vue + Vue Router + Vuex + Element UI 技术栈实现，项目打包通过 Webpack，代码转译通过 Babel。系统全局状态以及参数等信息通过 Vuex 进行保存，菜单图标采用 Icon Font，根据 BEM 命名规范编写全局通用组件样式，通过 Export2Excel 插件实现导出 excel 功能，与后端配合实现菜单权限，通过编写 Vue 指令实现操作按钮权限以及输入内容规则验证，通过原生 HTML、CSS、JS 封装上传文件、下载文件与预览图片等组件，统计图表通过 Echarts 展示效果。</p>		
2018.10-2019.3	深目系统	前端组成员
<p>该项目是为公安的公共安全方面提供服务的，全国在线下会布控很多摄像头，系统会在地图上显示出各个摄像头的位置等相关信息，摄像头会抓拍路人图片，然后上传到服务器，系统会分析这些图片中的人像生成相应的信息，还会把图片中分析出来的人像与公安系统中布控的人像图片进行比对，把人像相似度、行动轨迹等信息展示出来。</p> <p>该系统采用 React + React Router + Redux + Antd UI 技术栈实现，项目打包通过 Webpack，代码转译通过 Babel。系统全局状态以及参数等信息通过 Redux 进行保存，数据统计图表采用 Bizcharts 展示，系统中的地图用的是 Google Map，地图上不仅能够展示摄像头信息、行人图片等，还能够标记出被布控人员的行动轨迹。系统通过 MQTT 技术实现全局任务的发布与接收，通过 HTML 与 JS 的原生 API 实现图片的拖拽、保存、查询、任务发布等功能。</p>		