

黄柏祺

电话：18859595151 | 邮箱：huangboqi84@gmail.com
出生年月：2000年3月 | 年龄：25 | 工作经验：2年



教育经历

中南大学 - 通信工程 本科 2018年09月 - 2022年06月

工作经历

小米科技有限责任公司 - 软件研发工程师 2022年07月 - 2024年05月

互联网业务部·国际互联网部·研发平台·前端

- 主要负责小米国际应用商店开发者站的业务开发，独立完成多个功能模块的需求任务，并对其进行性能优化，提升前端体验和优化页面响应。
- 参与应用库、小米海外预装平台等基于 UmiJS 框架的后台系统的开发，进行公共方法的抽离以及通用组件的封装，高效完成多期功能需求。
- 参与翻译平台、FCM Push 平台等部门公共项目的开发，进行前端功能和 Node 服务的开发。
- 积极参与集团培训以及部门技术活动，完成基于Taro框架的微信小程序开发以及对网页版 GetApps 的无障碍适配。

专业技能

- 熟练掌握 JavaScript、HTML、CSS 等基础前端技术，熟练掌握 Promise、async/await 等 ES6+新特性和常用的弹性布局。
- 熟练掌握 Vue.js 框架以及 Vue Router、Vuex、Pinia 等相关生态技术，对其响应式原理、虚拟 DOM 等核心原理有较为深入的了解。
- 了解 React 框架技术的基本使用，具有基于 UmiJS 框架的开发经验。
- 了解 Node.js 和常用模块，具有基于 NestJS 框架的 Node 服务开发经验。
- 理解前端工程化思想，了解 webpack 基本使用，熟练使用 Git 进行版本控制及协同开发。

项目经历

FCM Push平台

- 项目描述**：该平台由一个基于 React 框架构建的平台页面以及一个基于 NestJS 框架构建的 Node 服务组成，核心功能是为部门多个业务产运人员提供创建、修改、审核以及发布 Push 消息推送的能力，由 Node 服务进行 Push 任务信息的各项处理并存储到数据库中。
- 技术选型**：React + Typescript + NestJS
- 工作内容**：
 - 完成主题管理、策略管理等提供增删改查能力的独立模块的开发，对现有任务创建能力进行扩展升级，接入部门翻译平台形成联动，实现一条任务最终可成为多条不同语言的任务的功能，提高产运人员的工作效率。
 - 使用 @nestjs/schedule 包设置定时任务，实现策略列表信息每日更新并且每日自动拉取部门内容平台内容池数据生成对应任务。
 - 封装透传消息键值对配置组件，包含删除现有配置项以及新增空白配置项的功能，在三种不同类型 Push 任务的创建页面中进行复用。
 - 基于 xlsx 库封装生成模板 Excel 文件的公共方法，此外实现了对已上传的 Excel 文件进行解析后批量生成对应内容的任务的功能。

项目经历

Mi Developer 小米国际应用商店开发者站

- **项目描述**：该项目是一个为小米 GetApp 应用开发者提供全方位服务的平台，涵盖开发者信息管理、应用发布与管理、订单管理、订阅服务以及账号权限管理等核心功能。
- **技术选型**：Vue + Vuex + ElementUI
- **工作内容**：
 - 进行用户权限、订阅管理、创收设置等多个复杂的全新功能模块开发以及现有功能模块的常规迭代。
 - 践行组件化思想，基于 ElementUI 组件库封装符合项目 UI 要求的对话弹窗、文本输入框、穿梭选择框等多个全局组件，提高代码复用率。
 - 设计并实现高阶组件 Authorized，在 render 函数中，根据鉴权逻辑结果动态决定渲染默认插槽内容或权限拦截页面，确保只有具备权限的用户才能访问特定页面，同时也减少了对现有页面代码的入侵性。
 - 灵活使用 mixin 功能，针对应用信息模块的多个页面进行应用详情获取、应用权限获取等公共方法以及路由守卫 beforeRouteLeave 的抽离复用，另外对 Vuex 的 commit & dispatch 以及 Vue 1.8 的 \$t 等高频方法作进一步封装，降低代码重复性，提高开发效率。
 - 初步尝试对项目进行性能优化，使用 Service Worker 对项目中 css & js 文件进行缓存，并使用 vue-lazyload 插件实现首页图片懒加载。

GetApps 应用管理平台

- **项目描述**：该项目是一个基于 UmiJS 构建的 CMS，核心功能是对收录在 GetApps 应用库中的应用进行管理，包含应用分级查询、开发者应用归属查询等五个自助工具以及应用收录、应用质检、应用监控等八个独立功能模块。
- **工作内容**：
 - 完成应用质检、功能标签管理、重点应用监控提醒等三个全新独立模块的开发以及其余五个现有模块的功能迭代。
 - 结合 File 和 FileReader API 实现分片上传 APK 包的功能。
 - 针对筛选栏中标签类型选择器 & 标签选择框对 useEffect 进一步封装，实现标签选择器中的标签选项与已选标签类型保持一致的功能。
 - 对 Ant Design 的树型选择组件进行二次封装，改造为可一键全选且可兼容不同数据结构的选项数组的选择器，在项目中 30+ 处进行复用，最小化筛选栏改版带来的代码变动。