

舒榆衡

男 | 年龄：25岁 | 17768938295 | shuyuheng@aliyun.com

6年工作经验 | 求职意向：全栈工程师 | 期望城市：北京



个人优势

5年全栈开发经验，后端功底扎实，具备Node.js（Nest/Koa）与Golang生产环境落地能力，擅长BFF层架构设计、高并发接口优化、数据库建模与缓存策略。曾驻场字节跳动（抖音）、滴滴等核心业务线，独立完成从0到1的后端服务设计与交付。

后端核心：Golang（接口服务、并发处理）、Node.js（BFF聚合、中间件）、MySQL（表设计/慢查询优化）、Redis（缓存/分布式锁）

架构能力：微服务调用（Thrift/Proto）、API网关聚合、规则引擎设计、热加载配置、RSA/AES加密

前端辅助：React 18 + TypeScript、Taro3跨端小程序（双端复用90%+），保障全栈闭环交付

工程化：CI/CD、GitFlow、Sentry监控、性能调优（接口响应从800ms→200ms）

工作经历

上海迪创网络科技有限公司（驻厂滴滴） 全栈工程师 2026.01-2026.06

内容：

后端架构：设计并实现CaseLint质量保障平台规则引擎，支持正则/模糊匹配/字段校验，规则热加载无需发版；基于Goroutine并发扫描+Redis缓存，将扫描耗时从5分钟/次降至2分钟/次（提升60%）。

数据库与接口：独立完成MySQL表设计、索引优化及Proto接口定义，统一前后端类型规范；开发Golang内网基建服务调用Node.js BFF层聚合多源数据，前端调用量减少40%。

全栈交付：主导Vue→React 18+TypeScript渐进式迁移，确保业务不中断；累计拦截30+线上风险，故障率降低10%，测试研发周期缩短1人/天。

业绩：

规则扫描速度从平均 5分钟/次 降至 2分钟/次，提升 60%

测试同学研发周期平均缩短 1人/天，提效约 20%

累计发现并拦截 30+ 个潜在线上风险，故障率降低约 10%

组件复用率提升 30%，新人上手成本显著降低

重庆易博互通科技有限公司（驻厂字节） 全栈工程师 2025.01-2026.01

内容：

后端服务：负责“可颂活动”小程序Golang服务层开发，对接MySQL/Redis实现活动报名、审核状态机及TCC下游数据导出；优化核心接口响应，服务可用性达99.9%+。

数据对接：集成飞书OpenAPI实现审批任务自动推送与结果回传，完成RSA/AES加密传输；设计多步骤表单草稿自动保存与恢复机制。

前端协同：基于Taro3封装动态表单与请求层，双端复用率90%+，支撑3轮体验迭代，报名转化率提升15%。

业绩：

线上问题响应缩短至 30 分钟内，服务可用性 99.9%+

完成 3+ 轮体验迭代，报名转化率提升 15%，次日留存提升 10%

需求交付周期缩短 1 人/天，迭代节奏从双周提速至周迭代

内容：

后端性能优化：Golang服务层对接MySQL、Redis，通过索引优化、连接池调整及批处理，核心接口响应从800ms降至200ms（提升75%）；Node.js BFF层聚合5+下游服务，前端代码量减少40%。

系统集成：飞书生态深度集成（在线文档、消息通知、审批流程），跨部门协作效率提升30%，审批流程缩短1人/天。

前端架构：基于React 18+Ant Design构建运营管理平台，通过懒加载、虚拟滚动将首屏从2.5s优化至1.2s，线上可用性维持在99.9%以上。

业绩：

首屏加载时间从 2.5s 优化至 1.2s（提升 52%）

核心接口响应从 800ms 降至 200ms（提升 75%）

飞书集成后跨部门协作效率提升 30%，审批流程缩短 1人/天

线上服务可用性维持在 99.9% 以上

项目经历

内容：

背景：QA 部门自动化 Case 及接口 API 缺乏规范检查，人工 Review 成本高。

职责：独立负责需求推进、全栈研发、上下游协调与交付。

技术：React 18 + TypeScript + Proto、Node.js/Golang、MySQL、Redis

工作内容：设计可配置规则引擎，支持正则/模糊匹配/字段校验，支持热加载，无需发版即可新增规则，基于 Proto 文件解析接口定义，自动校验字段命名、参数类型、返回值结构，替代人工 CodeReview 重复性检查

通过 Goroutine 并发扫描 + 规则缓存，扫描速度从 5分钟/次 降至 2分钟/次

完成 Vue→React 18 + TypeScript 渐进式迁移，按路由模块分批切换，业务不中断

业绩：

搭建可配置规则引擎，基于 Proto 解析接口自动校验，替代人工重复 CR，Review 工作量降低 60%

采用 Goroutine 并发 + 规则缓存优化，单次接口扫描由 5min 缩短至 2min，扫描效率提升 60%

完成 Vue 向 React18+TS 渐进式路由迁移，业务无中断，保障系统平稳迭代

内容：

背景：抖音旗下同城兴趣组局社交小程序，覆盖运动健身娱乐等垂类，需双端同步上线。

职责：负责活动创建、报名、审核、上下线完整闭环前端开发。

技术：Taro3 + React + TypeScript、Zustand、飞书 OpenAPI、RSA/AES Golang

工作内容：封装 Form/FormItem/ArrayField 动态表单，支持多场次/多规格/报名信息采集等复杂字段联动校验

设计活动状态机，统一管控草稿、审核、上下线等节点权限与页面跳转

对接飞书审批，审核任务自动推送，结果实时回传

使用 Zustand 管理多步骤表单，支持草稿自动保存与恢复

封装 useRequest、useLogin、useCountdown 等 Hooks 集群

业绩：

封装动态表单组件，完成复杂联动校验，减少表单类重复编码约 40%

设计活动状态机，统一管控多业务节点，状态类 BUG 下降 35%

对接飞书 OpenAPI 审批，实现任务自动推送、结果实时回传，审核流转效率提升 50%
基于 Zustand 实现草稿自动保存恢复，封装业务 Hooks 集群，页面开发周期缩短 30%

抖音-可颂运营平台（字节）

全栈开发工程师

2023.01-2025.06

内容：

背景：可颂运营平台前端工程及 API 服务，需构建强类型约束体系，保障代码可维护性与扩展性。

职责：负责前端工程搭建、API 服务开发、组件库沉淀。

技术：edexn + React + Gulu + TypeScript、Thrift、Axios、Golang、Mysql、Redis

工作内容：采用 Thrift 定义 API，自动生成 TS 类型文件，统一前后端数据类型规范，消除接口联调类型歧义

封装 Axios 实例，实现接口错误捕获、自动重试、多场景 code 码适配及 JWT 鉴权过期自动刷新

基于 useContext 做全局状态管理，结合 useMemo/React.memo/useCallback 优化组件渲染

封装表单、搜索组等通用组件并沉淀组件库，提升团队研发效率与页面风格一致性

业绩：

搭建可颂运营平台强类型体系，基于 Thrift 生成 TS 类型，接口联调问题下降 40%

封装 Axios 请求层，实现异常捕获、自动重试、JWT 刷新，接口故障率降低 35%

使用 React Hooks 优化组件渲染，减少无效重渲染

沉淀表单、搜索组通用组件库，团队开发效率提升 30%

抖音文娱管理平台 & 抖音运营平台（字节）

全栈开发

2021.08-2023.12

内容：

负责抖音文娱及运营管理平台全栈研发，覆盖媒资管理、评论审核、达人机构管理、数据看板等模块，应对高并发查询、复杂业务流转、多服务依赖等问题，兼顾服务稳定性与前端体验。

后端基于 Node.js 搭建 BFF 层聚合下游接口；使用 Golang 开发核心服务，优化 MySQL、Redis 查询与缓存策略，通过索引、批量处理提升接口吞吐，保障高并发稳定。

前端采用 React+TS+Semi Design 架构，Redux 管理复杂业务状态，封装通用 Hooks 与 Axios 请求层，做渲染、加载性能优化，迭代审核交互流程，减少误操作，提升业务开发与审核效率。

业绩：

后端基于 Node.js 搭建 BFF 层，减少前端对接代码约 35%；Golang 开发核心服务，优化 MySQL、Redis 缓存策略，核心接口响应耗时下降 40%，通过索引及批量处理改造，接口吞吐提升 50%，保障高并发下系统稳定。

前端采用 React+TS+Semi Design，Redux 管理业务状态；封装通用 Hooks 与请求层，组件复用率提升 45%，页面加载速度优化 30%；优化审核交互，操作误判率下降 28%，审核处理效率显著提升。

教育经历

上海开放大学

大专

计算机及应用

2018-2021