

程好

求职意向：开发工程师



学历：江西理工大学



生日：1996.07.03



电话：13016647040



邮箱：617240374@qq.com

## 专业能力

- 后端开发：精通 Python，熟练使用 Flask、Odoo、FastAPI、Django、Tornado 等框架进行项目开发；熟悉 Nginx、uWSGI、Gunicorn、aWSGI 等服务器配置与优化，熟悉爬虫，亚马逊 SP-API，了解 SPP 迁移，熟悉亚马逊 oauth2 实现以及密钥轮换和刷新，熟悉亚马逊 Lambda 方案实施。
- 数据库管理：熟练掌握设计 MySQL、PostgreSQL、MongoDB、Redis 等数据库的使用与管理，包括优化、迁移、备份、读写分离、分库分表策略，以及 DFS/MF、kafka 等分布式文件系统的应用。熟悉存储过程和能够通过动态接口设计批量设计近似表单查询，搭建简易中台。
- AI 工程应用落地：具备 AI 工程应用对接能力，能够将机器学习、深度学习等 AI 技术集成到现有系统中，提升系统的智能化水平。
- 高并发处理：具备处理海量用户并发及秒杀场景的设计、实现与部署经验，熟悉多种高并发技术体系，确保系统稳定高效运行。ES 等查询处理，以及常用 jieba 等分词处理。能够合理使用事务设计、多线程及锁设计，二次提交设计及令牌桶设计。
- 云原生技术：熟练掌握 Docker 与 Kubernetes 的部署与运维，推动应用向容器化、微服务化转型。
- 持续集成/持续部署：熟练使用 Git、Jenkins、SonarQube 等持续集成/持续交付（CI/CD）工具，提升开发效率和代码质量。曾从事代码质量检测，对代码质量和工程质量有保证。熟悉设计模式及算法结构。
- 数据分析与前端技术：熟悉使用 Requests、WebSocket、XPath、NumPy、Pandas 等工具进行数据收集与分析；具备 Vue.js、H5、CSS 等前端技术能力，能够设计并实现响应式、用户友好的 Web 界面。熟悉 uni-app 框架，拥有三端（iOS、Android、Web）程序开发经验，实现一次编写，多端运行。熟练使用 ECharts、ElementUI-plus 等主流前端库，提升前端开发效率和用户体验以及良好的组件封装能力。
- 额外技能：掌握 Go 语言、Rust、C++ 等编程语言基础，具备跨语言开发能力。
- 取得证书：华为 Python 专业级认证。

## 工作经历

## 江西泰豪股份有限公司（2016.06-2017.03） | 技术部 | PLC 设计工程师

- 培养了我对工业自动化流程的深刻理解，为后续的软件开发工作打下了坚实的基础。

## 深圳易新速科技有限公司（2017.03-2021.05） | 研发部 | 软件工程师

- 负责项目的全生命周期管理，包括独立开发、部署、接口文档的编写以及前后端的调试与维护。这一过程中，我熟练掌握了多种编程语言和技术栈，如 GO、Python 等，并成功应用于多个项目中。
- 小程序开发：我独立完成了多个小程序的开发与部署，从需求分析到界面设计，再到功能实现与测试，通过优化用户体验和性能，这些小程序在市场上获得了良好的反响。应对解决偶发问题。
- 数据分析与转化：在平安银行项目中，负责数据清洗、分析及转化工作，通过精细的数据处理，为银行提供了有价值的业务洞察，助力其决策优化。
- 团队指导与培养：作为团队中的资深成员，我积极指导和培养初级开发人员，通过分享经验、解答疑惑和提供实践机会，帮助他们快速成长，为团队注入了新的活力。

## 深圳德科有限公司 (2021.05-2023.07) | 2012 实验室开源能力中心 | 软件工程师 (D2)

- 开源代码检视工具扩展：利用 Python 和 GO 语言，基于 pylint、flake8 等流行的开源代码检视工具，编写了多条扩展规则。这些规则针对公司特定的编码规范和最佳实践进行定制，有效提升了内部代码库的质量和一致性。同时，我将这些扩展规则转化为内部使用的代码检视工具，实现了代码质量的自动化检测。
- 规则更新与内化定期跟进新规范和新技术的变化，为内部代码检视工具添加相应的规则，确保工具能够及时发现并纠正新出现的代码问题。加深对编程语言特性的理解，还提高了对新技术的敏感度。
- 复杂代码检视引擎开发：与南京大学合作期间，我参与了更为复杂的代码检视引擎的开发工作。该引擎不仅具备基本的代码漏洞检测能力，还能够实现跨模块的代码质量检视，为团队提供了更加全面和深入的代码分析服务。这一项目不仅锻炼了技术研发能力，增强团队合作精神和跨领域协作能力。

## 江西双胞胎集团 (2023.07-2024.04) | 流程与 IT 中心 | 全栈工程师

- Odoo 二次开发：根据业务需求，完成了多个 Odoo 模块的二次开发工作。从需求分析到设计方案制定，再到前后端开发及项目迭代，全程参与并确保项目的顺利进行。这些项目不仅提升了 erp 系统的功能性和用户体验，还为公司业务的发展提供了有力的支持。成本得到有效降低。
- PO 交互设计：为了实现 sap 软件和苍穹平台(养殖系统平台)之间的数据交互，承接设计和实现了 PO(Process Object) 交互方案。这一方案为两个系统之间的数据传输提供了稳定可靠的渠道，确保了数据的准确性和实时性。同时，参与了相关接口的开发和测试工作，确保了数据交互的顺畅进行。

## 个人开发 (2024.04-2024-11) | 智辉云 | 全栈

- 为 dd 德扑之家编写后台管理平台，控制 AI 加入坐下，以及通过对局条件做开始游戏离桌等决策。辅助算法，数据支持。

## 深圳开仓了科技有限公司 (2024.06-2025-06) | 软件开发部 | 开发工程师

- 灵狗退款：因核心开发离职接手。主要的业务节点如下：
  - 1、Bot 开发：通过 selenium 及商家子账号授权，拉取亚马逊的各订单核心信息，存储到 tidb 及灵狗后端数据库；
  - 2、灵狗退款平台：在 Django 框架下设计的一个 to C 软件，提供给商家用于统计自己的待退款单详情及分配业务员进行退款跟踪等核心业务。添加交易和赔偿单据追赔估价算法。
  - 3、灵狗退款平台重构业务：采用 seller 分离及单独的 seller 对应 sqlite，减少单一数据库的性能消耗，以用户为单位分表，降低成本。优化 bot 的查询逻辑，拓展数据上限承接更多的功能场景。避免使用高昂的亚马逊 tidb 服务。节约成本。尝试 swapon/异步/cdn 加速/多 PVC 等方式减低运行时间，增加稳定性。
  - 4、BOT 重构：基于以上的逻辑采用轮询令牌池发放对应的 seller\_id 进入对应 K8S 容器，跑完即停，将 seller 分类后放回池子底部，只需要处理各容器的生命周期，及时销毁和创造，达到 BOT 不会因为 IP 锁定导致系统性问题。
  - 5、编写仓储交易数据监控 demo。形成参考报表提供给决策层。
  - 6、K8S 部署以及处理内存问题，执行时间问题。
  - 7、与 PXW 开发编写 RPA 上位机控制识别测算材料 FBA 商品长宽/上位机指令动作处理
  - 8、提出和调研及实现通过亚马逊 Lamda 方案实现远程调用优化既有方案。减少下载的时间和网络传输及内存压力。

主要技术栈：Python、Selenium、PyQt、AwsClient、K8S、ES、Pandas、Mysql、Flask、Django、Sqlite

## 项目经历

## 深圳易新速科技有限公司 (2017.03-2019.05) | 太谷门户、利驰销售系统, 但眼考勤云

- 技术栈: Linux, FastAPI, MySQL, DFS (分布式文件系统), Elasticsearch, Kafka, RabbitMQ, Vue, ECharts, ElementUI。
- 项目简介: 在深圳易新速科技有限公司期间, 我参与了两个主要项目的开发工作。首先是为集团开发的太谷门户和利驰销售系统, 该系统主要用于地产移民及留学的资讯展示、引流, 以及返利秒杀、拉新返利等功能。其次, 我自研了钉钉同质项目——但眼考勤云, 该项目旨在实现考勤打卡、排班、请假等人力资源制度的全面数字化处理。
- 负责内容:**
- 但眼考勤云设计与开发:**
- 数据库与架构设计: 负责设计但眼考勤云的数据库结构, 确保数据的高效存储与查询。同时, 参与设计系统原型并完成逻辑梳理, 确保系统功能的合理性和可扩展性。
- 功能设计与实现: 完成了一系列核心功能的设计与开发, 包括排班即时性处理、定位信息校对、DFS 富文本排版显示、考勤通知推送、部门架构展示、权限设置、请假审批流程、人员变动与转岗管理、薪资调整机制、异常信息通知系统、日志埋点、用户访问跟踪以及智能补全搜索功能等。
- 搜索与实时通信: 部署并配置了 Elasticsearch, 实现了与 MySQL 的实时通信, 以支持搜索文本的智能补全功能, 提升了用户体验。
- 数据库与数据中台:**
- 设计并优化了数据库架构, 编写了存储过程和触发器以提高数据处理效率。
- 对 SQL 语句进行优化, 确保数据库查询的性能。
- 编写了动态接口, 支持系统的灵活扩展和定制化需求。
- 通过二次提交+令牌桶机制防止了超发问题, 保障了系统在高并发场景下的稳定性。
- 系统性能与安全:**
- 利用 Redis 订阅实现 Web 端和移动端的信息共享功能, 提高了系统的实时性和数据一致性。
- 通过二次提交+令牌桶机制防止了超发问题, 保障了系统在高并发场景下的稳定性。
- 前端与后端优化:**
- 完成了前端埋点工作, 用于收集用户行为数据, 为产品优化提供数据支持。
- 使用 Kafka 进行日志跟踪, 实现了系统的全面监控和故障排查。
- 实现了资讯分页功能, 并通过预热机制提升了资讯加载速度。
- 设计并优化了资讯热度排行算法, 确保了推荐内容的准确性和时效性。
- 部署与运维:**
- 负责项目的部署和运维工作, 确保系统能够正常发版和稳定运行。
- 监控系统性能, 及时发现并解决潜在问题, 保障系统的高可用性和用户体验。
- 管理和进度把控:**
- Apifox 及文档管理。周会进度。
- 方案实现建议。

## 深圳德科有限公司 (2021.05-2023.07) | 2012 实验室-可信实验室

- 技术栈: Pandas, Pylint, AST, Redis, Flake8, Go, Revive, XLSX, OpenCV, NumPy
- 项目简介: 在深圳德科有限公司的 2012 实验室-可信实验室项目中, 参与了公司代码质量的提升工作。通过制定代码规范、开发门禁工具以及建立案例场景库等措施, 有效提高了代码质量和开发效率。
- 负责内容:
- 开源工具比较与 AST 树解析: 比较并选用了多种开源代码检视工具, 并深入解析了 AST (抽象语法树) 以实现更精细的代码分析。通过算法实现, 我推断了部分标识符属性、函数参数异常等问题。
- 代码规则实现: 实现了一系列代码规则检测, 包括危险函数检测、死代码检测、返回类型不一致检测、无限循环检测、未释放资源检测以及返回个数设计不合理检测等。同时, 还实现了针对恶意拆分代码和强制类型转换等规避设计的负向整改规则。
- 数据分析与案例库建设: 通过分析代码清理反馈内容, 提炼了主要活跃部门、运营人员、有效问题数、有效问题链接及所属责任方等信息。这些信息被用于提炼意见收集群体、规范融合组扩编案例库以及有效问题所属规范等内容。
- 规则优化与用户答疑: 针对规则存疑的告警, 及时清理并为用户答疑。同时, 对存疑的规则进行了持续优化, 以提高检测的准确性和实用性。
- 门禁统计报表: 采集了各产品线未通过门禁代码的统计数据, 并输出了详细的报表, 为管理层提供了决策支持。

## 江西双胞胎集团 (2023.07-2024.06) | 苍穹-养殖系统平台

- 技术栈: Python, Odoo, HBuilderX, Vue, JavaScript, CSS, OpenXLSX, Framework7, RPC, Cordova
- 项目简介: 在江西双胞胎集团的苍穹-养殖系统平台项目中, 我主要负责了系统的数字化升级, 以减少养殖成本、避免不合规操作。通过数字化监控平台销售订单、物料领用、安全检测、存栏周期管理等各环节的业务审批及单据生成, 实现了养殖流程的透明化和智能化。
- 主要负责内容:
- Odoo 二次开发: 基于 Odoo 框架, 我开发了多个自定义页面和报表, 以满足养殖系统的特定需求。同时, 利用 WebSocket 技术实现了即时数据展示, 提高了数据的实时性和用户体验。
- PO 接口设计与控制: 使用 Python 的 request 库, 我设计了 PO (Process Object) 接口转接方案, 并编写了控制中心代码, 实现了不同系统之间的数据交互和流程控制。
- 数据库查询与存储过程: 利用 PostgreSQL 数据库, 我编写了多个查询语句和存储过程, 以支持复杂的业务逻辑和数据分析需求。解决慢查询问题。
- 前端开发与优化: 使用 Vue.js、JavaScript 和 CSS 等前端技术, 结合 HBuilderX 开发工具, 我编写了多个前端页面并修改了样式。同时, 针对 RPC 多层调用问题, 我建立了网关中心处理模块, 优化了系统架构, 避免了跨多级系统调用导致的发版问题。
- 脚本实现: workflows+markfiles 编写脚本自动项目打包, Pytest 测试用例编写以及 apipost 测试。

## 智辉云 (2024.04-2024.11) | DD 德扑之家

- 技术栈: Django, websocket, vue, Elementui-plus, ECharts, celery, redis, vue-count, mysql, MATLAB
- 项目简介: 控制 ai 落地操作及数据统计
- 负责内容:
- 平台开发: Apiview 进行开发, 图表开发。ai 规则逻辑实现等。
- 门禁统计报表: 采集了 AI 数据进行分析, 寻找有效影响胜率的因素。如卡方检测、回归性检测、分组数据 MATLAB



## 深圳 KCL 科技有限公司 (2024.05-2025.07) | 开发工程师

- 技术栈: Django, Redis, Pywright, sqlite, Aws Lambda, Selenium, Pygui, Request, Pandas, SQL 优化
- 项目简介: 多年爬虫项目的亚马逊订单及赔偿物流等数据过大, 导致许多慢查询已无法运转, 出于节约成本及不影响项目运行的考量, 架构师考虑转换服务, 并提出需要隔离 seller 的用户数据达到分治的作用, 因此启动此项目。
- **负责内容:**
- 将 tidb 所有的脚本做转化和优化 (含抓取报告, 存储 sqlite, 转化成业务数据, 定时清理和监控分析等)。(所有数据因条件不允许需要采用真实数据)。
- GMV 计算, 账单费率, 汇率, 收益统计及分析等
- 编写 AWS 索赔重试功能, 编写移除订单索赔功能。
- 页面更改随之带来的 bot 脚本更改。
- 运行出现的部分偶发性数据问题。
- RPA 运营, 编写 GUI 上位机, 提供单人操控机器及多流程动作统一处理以及上位机页面。
- 仓储交易数据监控及账单异常处理。
- 亚马逊 Lamda 方案实现和亚马逊 sp-api 更改及密钥轮换等问题
- 无测试环境开发, 主要依赖自测及线上真实数据操作。(出现不少问题, 但是细心程度直线上升)
- Sqlite 方案实施稳定后辞退, 计划奖金未发。曾表示节约一半以上服务器及数据库费用。