

个人简历

苏杭

男 | 4 年工作经验 | 26

现居住于北京

Tel: 150-0125-7313

E-mail: sul283034852@gmail.com

个人网站: <https://S1283034852.github.io>

教育经历

2011/09 -- 2015/07

宁夏师范学院 | 计算机科学与技术 | 统招本科

求职意向

工作性质: 全职

期望职业: Python 全栈开发工程师

工作地区: 北京

期望薪资: 面议

目前状况: 离职 (可立即到岗)

个人介绍

一直从事于 python 开发, 具有四年开发经验。Tornado 框架集成 celery, websocket 等第三方组件。熟练使用 Docker。了解微服务架构, 掌握 RPC 协议, 熟练使用 Thrift 框架。关于前端业务规范, 掌握前端框架 VUE, Element-ui 以及响应式开发框架 Bootstrap, 掌握 Mpvue 框架。掌握小程序、公众号开发。有优秀的学习能力和团队沟通能力, 经常与团队进行技术分享, 编写技术博客。

专业技能

1. 熟练使用 **python** 语言
2. 熟练应用 **Tronado, Django, Flask, Thrift** 等后端框架
3. 掌握 **VUE** 开发, 响应式开发框架 **Bootstrap, Flex, Mpvue**
4. 熟练使用 **Mysql, Redis, Mongo, Redisearch** 等数据库
5. 掌握 **Redis, Rabbitmq, Mamecach** 缓存和消息队列
6. 掌握微信小程序开发、公众号开发
7. 掌握 **Celery** 异步任务和定时任务
8. 掌握 **Git, Docker, Myqr** 的使用

工作经历

- 2016 年 10 月 - 至今 深圳依时货拉拉科技有限公司北京分公司一高级全栈开发工程师
- 2015 年 9 月 - 2016 年 9 月 北京纳谷科技有限公司一初级 python 开发工程师

项目经验

货车管理系统(货拉拉)

项目介绍:

顾名思义,该系统是用于货拉拉平台货车的管理。随着人流量的增大,注册我们的货车司机也与日俱增,我们放弃员工手动记录的古老办法,此系统为我们员工减少了大量的工作量。

技术要点:

- 1、基于 **Tornado** 实现 **Restful** 风格接口
- 2、采用 **Bootstrap** 响应式设计前端框架,方便管理人员随时审批和查看平台状态
- 3、使用 **websocket** 改造前端传统轮询技术框架,减少 30%的网络请求数,节约大约一半的可用带宽实现消息实时推送
- 4、采用 **Fstdfs** 分布式文件系统作为海量货车图片存储方案,并且结合 **FastDHT** 实现用户上传图片的分布式去重存储
- 5、**Nginx** 反向代理 Tornado,采用加权策略的负载均衡,后台服务统一使用 **SuperVisor** 进行管理
- 6、后期使用 **Thrift** 框架 **RPC** 协议架构对传统 http 接口进行重构,提高了整体接口的性能和吞吐量
- 7、使用 **Redisearch** 打造全文检索,百万级数据可以达到单次检索 10 毫秒以内的速度

货拉拉(微信小程序)

项目介绍:

货拉拉主要是一个 **O2O** 模式的在线叫车运货平台,类似于滴滴。当用户需要拉货搬家时,可以直接下载 app 或者微信小程序呼叫当地货拉拉司机。我们平台主要提供**同城/跨城**货运服务,平台货运车量涵盖从小面包车到 9 米 6 大货车多达 8 种车型,用户只需要通过小程序或 APP 一键呼叫,即可轻松预定下单附近的货车及司机,司机可以实时抢单并完成运输。

技术要点:

- 1、负责司机端后台用户的管理模块
- 2、完成货车类型选择,以及额外服务的展示、选择、下单功能
- 3、通过百度地图结合匈牙利算法实现附近车辆动态调度,为用户最快速度调度最近司机及货车资源

4、使用 **Redis** 作为缓存集群,redis sentinel 进行监控系统,司机同时抢单用 redis 分布式锁来完成派单

5、结合 **Celery** 完成用户下单成功、评价成功等场景下的消息推送

6、使用 **Supervisor** 对后台服务进行监控和操作,降低了日常维护成本

技术栈: Tornado、Mysql、JWT、Redis、Celery、RabbitMQ、Mpvue

好物平台

项目介绍:

好物是纳谷专门为用户推出的满足其生活消费和线上购物需求的平台,具有生活优惠信息、商品搜索、支付、收藏、订单查询、评论等在线功能,成为了用户方便快捷的生活消费入口。

个人职责: 用户认证、订单支付、消息推送、准时秒杀

技术要点:

1、项目整体采用前后端分离,接口使用 **drf** 实现,聚合支付宝、微信三方支付

2、使用 **Mysql** 存储数据,配置读写分离减轻数据库压力

3、利用 **websocket** 实现后端消息主动推送,改造前端传统轮询技术框架

4、使用 **Redis** 作为缓存介质,利用其分布式锁开发秒杀功能

5、部署采用 **Nginx+uwsgi** 的组合实现负载均衡