

# 吴翱翔的简历

17150313884 | me@tute.edu.cn

技术博客: [pymongo.github.io](https://pymongo.github.io/) | Github: [pymongo](https://github.com/pymongo)

求职意向: 全栈开发工程师 | 男/1997/本科

工作经历	海南自贸区威中杰信息技术有限公司	全栈开发	2019.10 - 至今
	· 参与数字货币交易平台、区块链节点的开发与维护, 主要负责WebSocket、数交所安卓端和聊天功能		
	北京邦友科技开发有限公司	软件测试	2018.11 - 2019.10
	· 主要业务是汽车ECU(电子控制器)的功能测试, 以日方提供的脚本语言进行编程, 调试		
贡献代码	sqlx开源项目	PR链接: <a href="https://github.com/launchbadge/sqlx/pull/319">https://github.com/launchbadge/sqlx/pull/319</a>	20.05.15 - 至今
	· Rust版sqlx与Go语言的sqlx类似, 是一个轻量的高性能的数据库工具 · 我阅读sqlx的MySQL源码时发现一处typo(单词拼写错误), 提交了PR修复了该文档拼写错误		
项目经验	杠杠交易引擎	项目链接: <a href="http://vue-trading.testcadae.top">http://vue-trading.testcadae.top</a>	20.05.26 - 至今
	· 基于Rust版币币交易撮合引擎, 添加杠杠交易的支持, 包含借币还币、做空做多等需求		
	Rust版币币撮合引擎	项目链接: <a href="http://vue-trading.testcadae.top">http://vue-trading.testcadae.top</a>	20.04.16 - 20.05.25
	· 耗时一个半月从零学习Rust语言, 一个人重写交易所撮合引擎, 技术栈是actix+sqlx · 相比ruby/sinatra旧版撮合, 性能从每秒处理25个下单请求提升到450个, 提升约20~25倍 · 作为项目负责人, 制定每日开发目标/TODO List, 编写项目文档, 提前交付项目 · 负责项目技术选型, 调研Go、Rust、C++等高性能语言, 项目中途将数据库框架更换为性能更好的sqlx · 攻克rust异步环境下多线程共享变量的死锁问题, 优雅地处理了Rust错误传递的过程 · 服务器架构上使用了两台redis一台MySQL, 使用Rust内存变量的方式缓存数据库的订单数据		
	数交所安卓APP	项目链接: <a href="http://app-download.cadex.top">http://app-download.cadex.top</a>	20.01.07 - 至今
	· 开发了OkHttpHelper, 为每个请求加上签名校验和设备信息(如机型, 安卓版本)等参数, 方便后端统计 · 实现了java websocket client, 包括断线重连、恢复订阅、取消全部订阅等功能 · 了解jetpack MVVM架构, 项目中用了的viewModel、DataBinding等新技术 · 将项目升级为androidx, 从安卓5.0兼容到安卓10, 了解dagger2依赖注入 · 通过Parcelable序列化/反序列化接口实现两个页面之间传递Java实例对象 · 从WebSocket获取数据的RecyclerView改为用SQLite的数据(Room ORM)进行初始化 · 集成了友盟SDK, 收集用户错误日志以及统计用户的使用信息, 方便产品运营		
	数交所安卓聊天功能		20.01.09 - 20.01.12
	· 数交所法币交易系统需要聊天功能让用户与商家取得联系, 安卓端的工作是对接现有WebSocket频道 · WebSocket客户端一开始用OkHTTP, 后来重构为java-websocket · 通过WebSocket心跳包技术实现了断线重连和保持连接功能 · 因服务端使用的MySQL版本过低无法存储Emoji表情包, 安卓聊天输入框设置了emoji过滤器		
	交易所移动端API接口	项目链接: <a href="http://v-api.testcadae.top/mobile_api/v2/otc">http://v-api.testcadae.top/mobile_api/v2/otc</a>	19.12.03 - 20.04.03
	· 践行TDD开发方式, 使用RESTful API框架grape, 通过单元测试/swagger/postman调试接口 · 给所有接口加上数字签名校验, 解决「N+1查询」问题, 通过ab命令测试接口性能 · 对于时效性短的数据, 存入redis减轻MySQL负载, 如当日取消订单3次限制交易的功能通过redis实现 · 使用FactoryBot库结合Faker库通过工厂模式生成测试数据, 借助单元测试帮组重构代码 · 在before :each的hooks将本次用到的相关数据表情况, 确保每个test case/example做到数据上的解耦		
	k线画线工具	项目链接: <a href="http://k-line-drawer.testcadae.top">http://k-line-drawer.testcadae.top</a>	19.11.18 - 19.12.01
	· 该项目是绘制币币交易系统部分对外展示的交易对的K线图的工具, 因机器人画线效果不佳需人工画K线 · 新增了两个接口, 一个用来给k线编辑器读取配置, 另一个是保存配置 · 独自完成了用户编辑完k线后, 提示有多少个柱子没画并且高亮这些柱子的功能 · 让element-ui的loading组件覆盖每个HTTP请求, 请求过程中把禁用按钮避免用户重复提交		
	每日利息与佣金计算脚本		19.11.14 - 19.11.23

- 该脚本计算DSCT矿池锁仓的每日利息以及用户推荐新人锁仓的佣金计算
- 用ruby元编程(define\_method)简化了佣金计算部分的代码, 通过「crontab」每日定时执行该脚本
- 学习建立User表与Account表之间的「双向关联」, 简化操作代码
- 在ActiveRecord的before\_update「hook/callback」给用户钱包余额加了非负验证
- 项目收获1: 资金应用要使用「Decimal」类型精准计算账户变动避免浮点数精度丢失
- 项目收获2: 涉及钱的操作要加「事务处理」和「线程锁/数据库锁」

交易所后台管理系统

项目链接: <http://gly.testcadae.top>

19.11.01 - 20.01.16

- 主要有运营、风控、用户提现审核等功能, 负责管理所有项目共用数据库的迁移文件
- 开发了用户支付方式、运营统计、数据图表、安卓版本管理等页面
- 新增了其它交易所数据爬虫机器人的管理的CRUD页面, 使用jQueryValidate进行表单验证
- 为所有导出csv下载的连接添加参数过滤(params.slice), 防止携带参数跳转时「无关的数据被暴露」
- 添加部分l18n英文和俄文的国际化支持, 通过参数merge解决了切换语言时GET请求参数被覆盖的问题

#### 个人作品

rust\_leetcode

项目链接: [https://github.com/pymongo/rust\\_leetcode](https://github.com/pymongo/rust_leetcode)

20.04.05 - 至今

- 使用Rust语言解LeetCode上的题, 独自思考, 通过之后才看别人的解答

go\_leetcode

项目链接: [https://github.com/pymongo/go\\_leetcode](https://github.com/pymongo/go_leetcode)

20.04.05 - 至今

- 由于Rust做算法题debug单步调试时经常跳进汇编代码中, 而Go语言不会, 所以先用Go解题

java\_leetcode

项目链接: [https://github.com/pymongo/java\\_leetcode](https://github.com/pymongo/java_leetcode)

20.04.05 - 至今

- 使用junit5 TDD开发leetcode的Java版解答

个人技术博客

项目链接: <https://pymongo.github.io>

18.08.16 - 至今

- 镜像: [wuaoxiang.github.io](http://wuaoxiang.github.io), 用Vue框架(SPA页面)做的个人博客, 整理总结学过的新技术

#### 技能清单

- **第一梯队(至少做过1个商业项目):** Rust(actix,diesel,sqlx)、Android、Ruby(rails,grape)
- **第二梯队(写过很多业余项目):** Python(flask)、JavaScript(Vue,jquery)、Java(junit)
- **第三梯队(做过leetcode,爱好者水平):** C/C++、Go(iris)、Arduino

声明: 本人承诺简历内容真实无误, 所有项目经历可在git上通过commit记录进行查询

致谢: 感谢您花时间阅读我的简历, 期待能有机会和您共事