

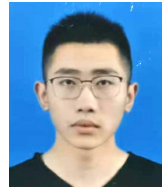
# 周靖

173-4334-9523 | zjing125.cn@gmail.com

武汉/杭州/Remote

Github : <https://github.com/jokereven>

WEB3开发/Go/Nodejs/ts/Python



## 教育经历

武汉职业技术学院 - 软件技术 大专

2020.10 - 2023.06

## 专业技能

- 1、掌握基础的 solidity 合约开发和部署，熟悉基于web3.js/web3.py/go-web3工具库的链上交互和链上事件/日志解析，熟悉 remixIDE/metamask等工具，对metaverse/NFT/DAPP有浓厚的兴趣和热情，了解NFT的各种玩法，了解BTC/ETH设计，InFuture成员，墙外玩家。
- 2、熟练使用 Gin/Koa 框架开发RESTful接口，熟悉 MySQL/Redis/MongoDB，能使用 React/Ts 开发web页面，掌握Go基础，了解goroutine/channel并发编程，了解go错误处理，垃圾回收，了解基于 Linux/Docker/Nginx/K8S 的前后端分离部署模式与简单运维，了解py爬虫。

## 项目经历

### 1、my-first-nft <https://github.com/jokereven/my-first-nft>

- 利用 hardhat 框架，开发并部署了一个基于以太坊智能合约的 NFT 项目，实现了图片的上传、铸造和交易功能
- 采用 ipfs 分布式存储技术，将图片数据保存在去中心化的网络上，保证了图片的安全性和唯一性
- 使用 Standard-Json-Input 格式编写合约元数据，并通过 solc 工具进行编译和验证，确保了合约代码的标准化和可靠性

### 2、链上数据解析

- 基于 alchemy 的 web3 节点，它可以通过指定合约/topic 来获取事件 event 的日志，并对解析出来的日志进行进一步的处理。然后，我使用 ts 对数据进行解码、过滤和格式化，最终实现了对以太坊上的交易日志的解析（部分交易所）。这个项目利用了 alchemy 提供的高可靠性、高可扩展性和高精度的 web3 API，以及合约/topic 的特性，可以快速地检索和分析区块链上的重要信息，为 dApp 开发者和用户提供更好的体验和服务。

### 3、win\_rate\_mint [https://github.com/jokereven/win\\_rate](https://github.com/jokereven/win_rate)

- 根据ETH上地址指定的GasFee/Value/mintNumber等条件，分析mint NFT的胜率 and 卖出胜率等，并将所有数据写入文件。
- 通过正则表达式批量筛选出符合预期的地址，然后通过py程序对这些地址进行Follow Mint：监听链上的交易，判断交易是否由筛选出的地址发起，如果是，就对交易进行解析，获取交易的类型和参数。如果交易是nft的mint交易，并且参数符合自己的承受范围，就进行模拟交易，评估gas费用。如果模拟交易成功，就发送真实交易。
- 通过tg推送Follow Mint的情况，同时使用discord对所有Follow地址的监控
- 链上实时监测mint情况，当达到预期的Mint情况进行推送。
- 对地址指定条件下NFT二级市场的胜率进行分析。

### 4、geth-cli <https://github.com/jokereven/geth-cli>

- 我使用go语言开发了一个简化版的比特币区块链系统，它涵盖了比特币的核心技术和概念，如区块链、交易、共识机制、数字签名等。我还设计了一个命令行界面，让用户可以方便地体验和探索各种功能。这个项目旨在深入理解和演示比特币的工作原理和实现细节。

### 5、基于 Agent/Kafka/Transfer/etcd 的海量日志收集/系统信息监控系统

- 实现了自定义日志收集服务组件和系统信息收集组件(Agent)，Agent 组件向 kafka 集群中发送消息数据，通过 Transfer 组件从 kafka 集群中消费数据并将日志数据异步转发给日志搜索分析系统、将系统信息数据写入时序数据库 influxDB，从

influxDB 中获取系统信息数据并做可视化展示以进行监控。

- 基于 goroutine 和 channel 实现自定义 LogAgent 和 SysAgent，从 etcd 配置中心读取配置，利用第三方库tail 从本机读取日志文件，利用第三方库 gopsutil 从本机获取系统信息，利用 channel 实现异步向 kafka集群中发送消息数据：Transfer 从 kafka 集群中消费数据，日志数据发往基于 Elasticsearch 和 Kibana 的可视化日志检索分析系统，系统信息数据写入 influxDB 时序数据库，并利用 Grafana 从 influxDB 中读取数据做可视化展示以进行系统监控。

## 6、基于 Gin/React 的内容社区网站后端/前端开发 Linux/Nginx/Docker 部署

- 后端实现用户注册登录及编辑管理等接口，文章上传/删除、获取列表/详情、点赞/点踩等接口，分类创建/删除/获取详情等接口，封装用户认证鉴权、网站限流等中间件，封装密码加密、错误信息翻译、ID生成等工具类库；管理后台实现数据浏览、文章审核等基础功能。
- 前端实现富文本编辑上传、图片裁剪上传、视频上传、未过期/未退出自动登录，用户注册登录，文章列表/详情，点赞/点踩，用户主页，个人信息编辑更新等功能与页面。
- 采用 CLD 分层管理代码，基于 viper 加载配置/zap 管理日志/snowflake 生成 ID/validator 校验参数/jwt 用户认证/token-bucket 限流管理等完成后端开发；使用 redis 数据库存储最新/最热文章 ID，以及存储文章实时浏览量、点赞数、综合得分等数据，分担 mysql 主数据库压力；完成数据库表、接口文档及代码注释撰写。
- 采用 React/Redux 技术栈，基于 react-router-dom 管理路由，styled-components 管理 CSS 样式，immutable/redux-thunk管理数据，axios 发送 ajax 请求，react-loadable 异步加载，antd、react-cropper 裁剪上传，braft-edit 实现富文本，localStorage 本地缓存完成前端开发。

## 个人总结

---

- 专长：编码 | 爱好：一切有趣的事
- 希望能在web3有自己的处女作，更相信Web3能让世界和平。