周靖

173-4334-9523 丨zjing125.cn@gmail.com

Wuhan/Hangzhou/Remote

Github：https://github.com/jokereven

WEB3 development/Go/Nodejs/ts/Python

EDUCATION

Wuhan vocational and technical college - Software technology Diploma 2020.10 - 2023.06

SKILLS LIST

1. Master the basic development and deployment of solidity contracts, be familiar with the link exchange and on-chain event/log analysis based on web3.js/web3.py/go-web3 tool library, be familiar with remixIDE/metamask and other tools, have strong interest and enthusiasm in metaverse/NFT/DAPP, understand various playing methods of NFT, understand BTC/ETH design, InFuture members and players outside the wall.

2. Proficient in using Gin/Koa framework to develop RESTful interfaces, familiar with MySQL/Redis/MongoDB, able to use React/Ts to develop web pages, master Go foundation, understand goroutine/channel concurrent programming, understand go error handling, garbage collection, understand front and rear separation deployment mode and simple operation and maintenance based on Linux/Docker/Nginx/K8S, and understand py crawler.

Project experience

1. my-first-nft https://github.com/jokereven/my-first-nft

- Using the hardhat framework, we developed and deployed an NFT project based on Ethernet Smart Contract, which realized picture upload, casting and trading functions

  .
- The ipfs distributed storage technology is used to store image data on a decentralized network, ensuring the security and uniqueness of images.
- Use Standard-Json-Input format to write contract metadata, and use solc tools to compile and verify, ensuring contract code standardization and reliability

2. Data analysis on the chain

- is a alchemy-based web3 node, which can obtain event logs by specifying contracts/topics and further process the parsed logs. Then, I used ts to decode, filter and format the data, and finally realized the analysis of the transaction log on the Ethernet Square (some exchanges). This project takes advantage of the high reliability, high scalability and high accuracy web3 API provided by alchemy, as well as the characteristics of contract/topic, which can quickly retrieve and analyze important information on the block chain and provide dApp developers and users with better experience and services.

3. win_rate_mint https://github.com/jokereven/win_rate

- According to the ETH address specified GasFee/Value/mintNumber and other conditions, analyze mint NFT win rate and sell win rate, etc., and write all data to the file.
- uses regular expressions to batch filter out addresses that meet expectations, and then uses the py program to Follow Mint: monitor transactions on the chain to determine whether the transaction was initiated by the filtered address, and if so, parse the transaction to obtain the type and parameters of the transaction. If the transaction is a mint transaction of nft and the parameters meet your tolerance range, conduct a simulated transaction and evaluate the gas fee. If the simulated transaction is successful, send the real transaction.
- Through tg push Follow Mint situation, while using discord for all Follow address monitoring

  The

- chain real-time monitoring mint situation, when the expected Mint situation is pushed.
- The address specified conditions NFT secondary market win rate analysis.

4. geth-cli https://github.com/jokereven/geth-cli

- I have developed a simplified version of bitcoin block chain system using go language, which covers the core technologies and concepts of bitcoin, such as block chain, transaction, consensus mechanism, digital signature, etc. I also designed a command line interface that allows users to easily experience and explore various functions. This project aims to deeply understand and demonstrate the working principle and implementation details of bitcoin.

5. Massive log collection/system information monitoring system based on Agent/Kafka/Transfer/etcd

The

- implementation of the custom log collection service component and system information collection component (Agent), Agent component to the kafka cluster to send message data, through the Transfer component from the kafka cluster consumption data and log data asynchronous forwarding to the log search analysis system, the system information data written to the time series database influxDB, from the influxDB to obtain system information data and do visual display for monitoring.
- implements custom LogAgent and SysAgent based on goroutine and channel, reads the configuration from etcd configuration center, uses the third-party library tail to read log files from this machine, uses the third-party library gopsutil to obtain system information from this machine, and uses channel to realize asynchronous sending of message data to kafka cluster: Transfer consumes data from kafka cluster, and the log data is sent to the visual log retrieval and analysis system based on ElasticSearch and Kibana, system information data is written into the influxDB time series database, and data is read from the influxDB by the Grafana for visual display for system monitoring.

6. Linux/Nginx/Docker deployment based on Gin/React content community website back-end/front-end development

- The back-end implements interfaces such as user registration, login and editing management, interfaces such as article upload/delete, list/details acquisition, praise/click, etc., interfaces such as classification creation/deletion/details acquisition, middleware such as user authentication and authentication, website current limiting, and tool libraries such as password encryption, error information translation, ID generation, etc.; The management background realizes basic functions such as data browsing and article review.
- The front end realizes functions and pages such as rich text editing and uploading, picture clipping and uploading, video uploading, automatic login without expiration/exit, user registration and login, article list/details, praise/click, user homepage, personal information editing and update, etc.
- CLD hierarchical management code is adopted to complete back-end development based on viper loading configuration/zap management log/snowflake generation ID/validator verification parameter/jwt user authentication/token -bucket current limiting management, etc. Redis database is used to store the latest/hottest article ID, as well as the real-time page views, praise numbers, comprehensive scores and other data of articles, sharing mysql main database pressure; Complete database table, interface document and code annotation writing.
- React/Redux technology stack is adopted, routing is managed based on react-router-dom, CSS style is managed by styled-components, data is managed by immutable/redux-thunk, ajax request is sent by axios, react-loadable is loaded asynchronously, antd and react-cropper are cropped and uploaded, rich text is implemented by draft-edit, and front-end development is completed localstorage local cache.

SUMMARY

- Expertise: Coding | Hobbies: Everything Interesting
- I hope I can have my first work in web3, and I believe that Web3 can make world peace.