Contents

[**I.** **Overview** 2](#_Toc183890581)

[**II.** **Technical stack using** 2](#_Toc183890582)

[**III.** **Design patterns list** 2](#_Toc183890583)

[**IV.** **Utilities** 2](#_Toc183890584)

[**V.** **Database model** 3](#_Toc183890585)

[**1.** **MARKET\_API** 3](#_Toc183890586)

[**2.** **CURRENCY\_LIST** 3](#_Toc183890587)

[**3.** **CRON\_SCHEDULE** 3](#_Toc183890588)

[**4.** **CRON\_JOB** 4](#_Toc183890589)

[**5.** **BITCOIN\_EXCHANGE** 4](#_Toc183890590)

[**6.** **BITCOIN\_ANALYZE** 4](#_Toc183890591)

[**VI.** **Function list** 4](#_Toc183890592)

[**1.** **Simulate bitcoin price data, import into CAB database with Flyway** 4](#_Toc183890593)

[**2.** **Process bitcoin price analysis from Flyway data and insert to BITCOIN\_ANALYZE table, when CAB app is started** 5](#_Toc183890594)

[**3.** **View marketApi list** 5](#_Toc183890595)

[**4.** **Crawl bitcoin price coindesk API, binance API, and other market API by manual** 5](#_Toc183890596)

[**5.** **Crawl bitcoin price coindesk API, binance API, and other market API by cronJob schedule** 5](#_Toc183890597)

[**6.** **Convert the data from coindesk API, binance API, and other market API and store into BITCOIN\_** **EXCHANGE table** 5](#_Toc183890598)

[**7.** **View bitcoin price data crawled in CAB database** 6](#_Toc183890599)

[**8.** **Process Bitcoin Price Analyze Manually Daily/Weekly/Monthly** 6](#_Toc183890600)

[**9.** **Process Bitcoin Price Analyze Daily/Weekly/Monthly by cronJob schedule** 6](#_Toc183890601)

[**10.** **Process Bitcoin Price Analyze Manually Daily/Weekly/Monthly with encryption and decryption technology (AES)** 7](#_Toc183890602)

[**11.** **View analyzed bitcoin price of current day and previous days** 8](#_Toc183890603)

[**12.** **View analyzed bitcoin price of previous days/weeks/months** 8](#_Toc183890604)

[**13.** **View list cronJob schedule called to crawl API bitcoin price** 8](#_Toc183890605)

[**14.** **View list cronJob schedule analyze bitcoin price** 8](#_Toc183890606)

[15. **Print out the request and response body log of all API internal and external**. 9](#_Toc183890607)

[**16.** **Error handling to decorate all API responses** 9](#_Toc183890608)

1. **Overview**

The crawler and analyze bitcoin price tool, I call it **CAB** tool for short.

Bitcoin is growing, and the number of bitcoin users is huge.

There are increasingly more bitcoin exchanges, and the exchange rates and value of bitcoin are constantly rising. To collect and analyze bitcoin prices on various bitcoin exchanges, as well as compare bitcoin prices between exchanges, the CAB tool undertakes the mission of collecting and analyzing bitcoin prices on exchanges such as Coindesk, Binance, etc.

Data is collected and analyzed in real-time. Users can view analyzed data on a daily, weekly, or monthly basis, including: opening price, closing price, highest price, lowest price, percentage change.

The CAB tool's bitcoin price analysis model is referenced on the website:

<https://www.investing.com/crypto/bitcoin>

<https://www.investing.com/crypto/bitcoin/historical-data>

1. **Technical stack using**

|  |  |  |  |
| --- | --- | --- | --- |
| Spring-boot v3.4.0 | Java: 17 | Swagger | Flyway |
| Repository: H2 (OpenJPA/Spring Data JPA) | encryption and decryption  with AES | Quartz Schedule | Junit |

1. **Design patterns list**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Factory patterns | Abtract Factory patterns | Strategy patterns | Singleton patterns | Adapter pattern | Builder pattern |

1. **Utilities**

|  |  |
| --- | --- |
| H2 database url | <http://localhost:8080/h2-console>  JDBC url: jdbc:h2:mem:analyzedb  User name: sa |
| Swagger url | http://localhost:8080/swagger-ui/index.html |
| Postman file | bitcoin-price.postman\_collection.json |
| Sql query | sql query.txt |

1. **Database model**

A diagram of a company

Description automatically generated with medium confidence

Data information example in table database:

1. **MARKET\_API**



1. **CURRENCY\_LIST**

A screenshot of a computer

Description automatically generated

1. **CRON\_SCHEDULE**

A screenshot of a computer

Description automatically generated

1. **CRON\_JOB**

A screenshot of a computer

Description automatically generated

1. **BITCOIN\_EXCHANGE**

A table with numbers and symbols

Description automatically generated

1. **BITCOIN\_ANALYZE**

A screenshot of a computer

Description automatically generated

1. **Function list**
2. **Simulate bitcoin price data, import into CAB database with Flyway**

* File Flyway destinations:

src\main\resources\db\migration\V1\_\_init\_database.sql

src\main\resources\db\migration\V2\_\_init\_Bitcoin-Exchange-data-history.sql

* File excel destinations:

init\_Bitcoin-Exchange-data-history.xlsx

1. **Process bitcoin price analysis from Flyway data and insert to BITCOIN\_ANALYZE table, when CAB app is started**

* File destinations:

src\main\java\myapp\bitcoin\_analyze\service\impl\BitcoinAnalyzeServiceImpl.java\initBitcoinAnalyze

1. **View marketApi list**

* Postman:

Method Get : <http://localhost:8080/api/market-apis?&sort=id,desc&page=0&size=10>

* File destinations:

src\main\java\myapp\bitcoin\_analyze\web\rest\MarketApiResource.java\getAllMarketApis

1. **Crawl bitcoin price coindesk API, binance API, and other market bitcoin API by manual**

* Postman:

Method Get : <http://localhost:8080/api/bitcoin-exchanges/fetchCurrentPrice>

* File destinations:

src\main\java\myapp\bitcoin\_analyze\web\rest\BitcoinExchangeResource.java\fetchCurrentPrice

1. **Crawl bitcoin price coindesk API, binance API, and other market bitcoin API by cronJob schedule**

* File destinations:

src\main\java\myapp\bitcoin\_analyze\service\impl\ScheduleServiceImpl.java\init

src\main\java\myapp\bitcoin\_analyze\scheduler\ ScheduleFetchCurrentPrice.java\executeSchedule

src\main\java\myapp\bitcoin\_analyze\service\impl\ScheduleServiceImpl.java\ scheduleFetchCurrentPrice

1. **Convert the data from coindesk API, binance API, and other market bitcoin and store into BITCOIN\_** **EXCHANGE table**

* File destinations:

src\main\java\myapp\bitcoin\_analyze\service\CoinDeskService.java\fetchCurrentPrice

src\main\java\myapp\bitcoin\_analyze\service\BinanceService.java\fetchCurrentPrice

1. **View bitcoin price data crawled in CAB database**

* Postman

Method Get : <http://localhost:8080/api/bitcoin-exchanges?sort=createdDate,desc&page=0&size=10>

* File destinations:

src\main\java\myapp\bitcoin\_analyze\web\rest\BitcoinExchangeResource.java\ getAllBitcoinExchanges

1. **Process Bitcoin Price Analyze Manually Daily/Weekly/Monthly**

* Postman

Method Post : <http://localhost:8080/api/bitcoin-analyzes/process-analyze>

* Request body:

|  |  |  |  |
| --- | --- | --- | --- |
| Analyze bitcoin price at previous day | Analyze bitcoin price at previous week | Analyze bitcoin price at previous month | Analyze bitcoin price at previous day, week, month |
| {"timeType":"DAILY"} | {"timeType":"WEEKLY"} | {"timeType":"MONTHLY"} | empty |

* File destinations:

src\main\java\myapp\bitcoin\_analyze\web\rest\BitcoinAnalyzeResource.java\ processBitcoinAnalyzeManual

1. **Process Bitcoin Price Analyze Daily/Weekly/Monthly by cronJob schedule**

* File destinations:

+ Previous day:

src\main\java\myapp\bitcoin\_analyze\service\impl\ScheduleServiceImpl.java\init

src\main\java\myapp\bitcoin\_analyze\scheduler\ ScheduleBitcoinAnalyzePreviousDay.java\executeSchedule

src\main\java\myapp\bitcoin\_analyze\service\impl\ScheduleServiceImpl.java\ ScheduleBitcoinAnalyzePreviousDay

+ Previous week:

src\main\java\myapp\bitcoin\_analyze\service\impl\ScheduleServiceImpl.java\init

src\main\java\myapp\bitcoin\_analyze\scheduler\ ScheduleBitcoinAnalyzePreviousWeek.java\executeSchedule

src\main\java\myapp\bitcoin\_analyze\service\impl\ScheduleServiceImpl.java\ ScheduleBitcoinAnalyzePreviousWeek

+ Previous month:

src\main\java\myapp\bitcoin\_analyze\service\impl\ScheduleServiceImpl.java\init

src\main\java\myapp\bitcoin\_analyze\scheduler\ ScheduleBitcoinAnalyzePreviousMonth.java\executeSchedule

src\main\java\myapp\bitcoin\_analyze\service\impl\ScheduleServiceImpl.java\ ScheduleBitcoinAnalyzePreviousMonth

1. **Process Bitcoin Price Analyze Manually Daily/Weekly/Monthly with encryption and decryption technology (AES)**

* Postman:

Method Post : <http://localhost:8080/api/bitcoin-analyzes/process-analyze-aes>

* Request body:

|  |  |  |  |
| --- | --- | --- | --- |
| Analyze bitcoin price at previous day | Analyze bitcoin price at previous week | Analyze bitcoin price at previous month | Analyze bitcoin price at previous day, week, month |
| {      "requestDate":"2024-11-25",      "requestTime":"17:19:00",      "timeType":"DAILY"  } | {      "requestDate":"2024-11-25",      "requestTime":"17:19:00",      "timeType":" WEEKLY"  } | {      "requestDate":"2024-11-25",      "requestTime":"17:19:00",      "timeType":" MONTHLY"  } | {      "requestDate":"2024-11-25",      "requestTime":"17:19:00"  } |

* Script > Pre-quest:

|  |
| --- |
| **var** CryptoJS = require(“crypto-js”);  **var** moment = require(“moment”);  **var** myReq = JSON.parse(pm.request.body.raw);  **var** CryptoJS = require(“crypto-js”)  **let** requestDate,requestTime;  requestDate = myReq.requestDate;  requestTime= myReq.requestTime;  **var** aesSecretKey = “OxLDVPTHLk5EHR5AE8O0rg==”;  aesEncrypt(aesSecretKey);  **function** aesEncrypt(aesSecretKey) {  **const** method = pm.request.method;  **const** data = `${method}|${requestDate}|${requestTime}|${JSON.stringify(myReq)}`  **const** encryptedText = CryptoJS.AES.encrypt(data, aesSecretKey);      pm.request.headers.add({          key: ‘signature’,          value: `${encryptedText}`      });  } |

1. **View analyzed bitcoin price of current day and previous days**

* Postman:

Method Get : <http://localhost:8080/api/bitcoin-analyzes/general-data?fromDate=2024-11-22&currencyCode=USD&marketId=coindesk&timeType=DAILY&sort=id,desc&page=0&size=20>

* File destinations:

src\main\java\myapp\bitcoin\_analyze\web\rest\BitcoinAnalyzeResource.java\getAllGeneralData

1. **View analyzed bitcoin price of previous days/weeks/months**

* Postman:

Method Get : <http://localhost:8080/api/bitcoin-analyzes/historical-data?fromDate=2024-11-25&toDate=2024-11-26&currencyCode=USD&marketId=coindesk&timeType=DAILY&sort=id,desc&page=0&size=10>

* timeType=DAILY
* timeType=WEEKLY
* timeType=MONTHLY
* File destinations:

src\main\java\myapp\bitcoin\_analyze\web\rest\BitcoinAnalyzeResource.java\getAllHistoricalData

1. **View list cronJob schedule called to crawl API bitcoin price**

* Postman:

Method Get : <http://localhost:8080/api/cron-jobs?jobName=FetchCurrentPrice&startTime=2024-11-29T00:00:00Z&endTime=2024-11-30T00:00:00Z&sort=id,desc&page=0&size=10>

* File destinations:

src\main\java\myapp\bitcoin\_analyze\web\rest\CronJobResource.java\getAllCronJobs

1. **View list cronJob schedule analyze bitcoin price**

* Postman

Method Get : <http://localhost:8080/api/cron-jobs?jobName=BitcoinAnalyze&startTime=2024-11-29T00:00:00Z&endTime=2024-11-30T00:00:00Z&sort=id,desc&page=0&size=10>

* File destinations:

src\main\java\myapp\bitcoin\_analyze\web\rest\CronJobResource.java\getAllCronJobs

1. **Print out the request and response body log of all API be called and call out external APIs**.

* File destinations:

src\main\java\myapp\bitcoin\_analyze\config\LoggingInterceptor.java

src\main\java\myapp\bitcoin\_analyze\config\RestTemplateLoggingInterceptor.java

1. **Error handling to decorate all API responses**

* File destinations:

src\main\java\myapp\bitcoin\_analyze\service\ValidateService.java