**금융 IT 개발 실무 경험하기**

**코멘토 연동규격서 작성**



Document Information

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: | Comento 금융IT 개발 실무 | | |
| Project Manager: | 리드멘토/또또 | Version Number: | V0.1 |
|  |  | Version Date: | 2021.09.25 |
| Author: | 코멘토/리드멘토 | Written Date: | 2021.09.25 |
| Reviewer: |  | Reviewed Date: | 2021.09.25 |

Document Distribution

| Copy Number | Name(Role, Title) | Date | Contact (Phone/email) |
| --- | --- | --- | --- |
| 1 | 리드멘토 | 2021.09.25 | 010-0000-0000  abc@naver.com |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Revision History

| Version | Version Date | Author | Description |
| --- | --- | --- | --- |
| 0.1 | 2021.09.07 | 코멘토/리드멘토 | 초안 작성. |
| 0.2 | 2021.09.09 | 코멘토/리드멘토 | 연동 API 추가 |
| 0.3 | 2021.09.15 | 코멘토/리드멘토 | API(스크래핑)정보 추가 |
| 0.4 | 2021.09.25 | 코멘토/리드멘토 | 샘플추가 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. 개요 4
2. 목적 4
3. 범위 4
4. 용어 정의, 약자, 약어 4
5. 참고 문서 4
6. Rest-ful API 연동 규격서 소개 5
7. 참조 내용 5
8. 연동규격 내용 6
9. 연동 방식 6
10. 서버정보 6
11. 연동상세 6
12. Rest-ful API 연동규격 7
13. 해당종목의 테마 이름 7
14. 코스피/코스닥 상위 5종목 시총 합 8
15. 코스피/코스닥 시총 상위 30개 종목코드 및 종목 명 9
16. 종목코드로 기업 종합 정보 조회 10
17. 코스닥 종목중 가장 높은 ROE를 가진 종목의 정보를 조회 11
18. Appendix 암호화/복호화 방식 참고(java 기준) 12
19. 추가 논의 사항 13
20. 개요
21. 목적

본 문서의 목적은 다음과 같다.

이 문서는 comento 멘토링에서 제공하는 Rest-ful API연동에 관한 내용을 정의한다.

금융 IT 개발 실무를 경험하기위해 만들어진 문서이다.

1. 범위

본 문서의 범위는 다음과 같다.

본 문서에서는 금융 IT 개발 실무 내용을 Rest-ful API 관련 정보 구성과 사용법에 대해 설명하고 있다.

1. 용어 정의, 약자, 약어

|  |  |
| --- | --- |
| 용 어 | 내 용 |
| HTTP | Hypertext Transfer Protocol. |
| HTTPS | Hypertext Transfer Protocol over Secure Socket Layer. |
|  |  |

1. 참고 문서

없음

1. Rest-ful API 연동 규격서 소개

이 문서는 금융 IT 개발 실무에서 1주차, 2주차에서 멘토링한 내용을 기반으로 요구사항에 맞는 연동규격서 내용으로 업체에 대한 정보를 제공한다.

1. 참조 내용

Rest-ful API 연동 규격서에 포함된 단원은 다음과 같다 . 각 함수는 다음과 같은 정보를 가지고 있다.

|  |  |
| --- | --- |
| 항목 | 상세 |
| 데이터 구조 | 사용자 인증 (기본타입) API 에서 사용되는 데이터 구조를 사전 순으로 나열한다. |
| 프로토타입 | 호출 구조에 대한 예제이다. |
| 매개 변수 | 입력되어야 할 항목과 반환되는 항목이다. |
| 설명 | 함수 사용에 대한 설명이다. |
| 반환 값 | 함수의 호출에서 반환되는 다양한 유형, 메시지, 값, 구조 및 설명을 포함하고 있는 항목이다. |
| 설명 | 함수의 사용, 한계 및 영역에 대한 이해를 돕기 위한 특별 설명과 부가 정보가 있다. |

1. 연동규격 내용
2. 연동 방식

A. 서버 to 서버

- 연동 프로토콜은 HTTPS 기본으로 한다.(상용 : 필수, 개발/QA : 옵션)

* + - 주요 정보는 암/복호화 연동 처리 한다.
    - 연동 KEY :

[개발] : abc1234567891234(16자리)

[스테이징] : abc1234567891234(16자리)

[상용] : <별도 전달>

* + - 암/복호화 KEY :

[개발] key : abc1234567891234(16자리)

iv : abc1234567891234(16자리)

[스테이징] : key : abc1234567891234(16자리)

iv : abc1234567891234(16자리)

[상용] : <별도 전달>

B. 암/복호화 방식

- AES128

1. 서버정보

A. 서버

[개발] : https://dev.comento.co.kr

[스테이징] : https://dev.comento.co.kr

[상용] : <https://comento.co.kr>

B. 포트

개발 : 5321

스테이징 : 5322

운영 : 8855

1. 연동상세

| **구분** | **설명** |
| --- | --- |
| 문자 인코딩 | UTF-8을 기본으로 한다. |
| 서버 요청 | Content-type : application/x-www-form-urlencoded; charset=UTF-8 |
| 서버 응답 | Content-type : application/json; charset=UTF-8 |
| 파라미터 | 파라미터는 대소문자를 구분한다. |
| 길이 | Byte 기준 |
| 필수 | M : 필수항목  O : 선택항목 |

1. Rest-ful API 연동규격

\* 정의 주체 : 코멘토

\* 사용 주체 : 업체

1. 해당종목의 테마 이름

해당종목의 테마이름 조회

|  |  |
| --- | --- |
| URL | /comento/biz/code/theme |
| Content-Type | application/x-www-form-urlencoded; charset=UTF-8 |
| Request Method | GET |

**- Request body**

|  |  |  |  |
| --- | --- | --- | --- |
| **파라미터명** | **타입** | **필수여부** | **설명** |
| code | String | **Y** | 종목코드 |
|  |  |  |  |
|  |  |  |  |

* **Response body**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **Depth** | **배열**  **구분** | **설명** | **값 구분** |
| result | 1 |  | 해당 아이디어에 상태 업데이트에 대한 결과 값 | sucess : 성공(정상)  error : 실패 |
| description | 1 |  | 결과(result 엘리먼트)에 대한 설명 |  |
| theme\_name | 1 |  | 테마이름 |  |

**\*** 샘플 JSON예제

|  |
| --- |
| - 실패할 경우 샘플  {“result":"error","description":"etc"}  -성공할 경우 샘플  {“result”:”sucess”,”description”:”sucess”,"theme\_name":"반도체\_생산"} |

1. 코스피/코스닥 상위 5종목 시총 합

코스피/코스닥 상위 시총 합 조회

|  |  |
| --- | --- |
| URL | /comento/biz/marketcap/sum |
| Content-Type | application/x-www-form-urlencoded; charset=UTF-8 |
| Request Method | GET |

**- Request body**

|  |  |  |  |
| --- | --- | --- | --- |
| **파라미터명** | **타입** | **필수여부** | **설명** |
| market | String | **Y** | market(ex: kospi, kosdaq) |
|  |  |  |  |
|  |  |  |  |

* **Response body**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **Depth** | **배열**  **구분** | **설명** | **값 구분** |
| result | 1 |  | 해당 아이디어에 상태 업데이트에 대한 결과 값 | sucess : 성공(정상)  error : 실패 |
| description | 1 |  | 결과(result 엘리먼트)에 대한 설명 |  |
| marketCapSum | 1 |  | 상위 5종목 시총 합(코스피 or 코스닥) |  |

**\*** 샘플 JSON예제

|  |
| --- |
| - 실패할 경우 샘플  {“result":"error","description":"etc"}  -성공할 경우 샘플  {“result”:”sucess”,”description”:”sucess”,”marketCapSum":"3929161"} |

1. 코스피/코스닥 시총 상위 30개 종목코드 및 종목 명

코스피/코스닥 시총 상위 30개 종목코드 및 종목명

|  |  |
| --- | --- |
| URL | /comento/biz/code/30 |
| Content-Type | application/x-www-form-urlencoded; charset=UTF-8 |
| Request Method | GET |

**- Request body**

|  |  |  |  |
| --- | --- | --- | --- |
| **파라미터명** | **타입** | **필수여부** | **설명** |
| market | String | **Y** | market(ex: kospi, kosdaq) |
|  |  |  |  |
|  |  |  |  |

* **Response body**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **Depth** | **배열**  **구분** | **설명** | **값 구분** |
| result | 1 |  | 해당 아이디어에 상태 업데이트에 대한 결과 값 | sucess : 성공(정상)  error : 실패 |
| description | 1 |  | 결과(result 엘리먼트)에 대한 설명 |  |
| codeList | 1 |  | codeList |  |
| code | 2 |  | 종목코드 |  |
| code\_name | 2 |  | 종목명 |  |

**\*** 샘플 JSON예제

|  |
| --- |
| - 실패할 경우 샘플  {“result":"error","description":"etc"}  -성공할 경우 샘플  {“result”:”sucess”,”description”:”sucess”,”codeList”: [  {“code” : “1234567”, “code\_name” : “삼성전자”},  {“code” : “1234568”, “code\_name” : “하이닉스”}  ]  } |

1. 종목코드로 기업 종합 정보 조회

종목코드로 기업 종합 정보 조회

|  |  |
| --- | --- |
| URL | /comento/biz/code/detail |
| Content-Type | application/x-www-form-urlencoded; charset=UTF-8 |
| Request Method | GET |

**- Request body**

|  |  |  |  |
| --- | --- | --- | --- |
| **파라미터명** | **타입** | **필수여부** | **설명** |
| code | String | **Y** | 종목코드 |
| market | String | **Y** | market(ex: kospi, kosdaq) |
|  |  |  |  |

* **Response body**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **Depth** | **배열**  **구분** | **설명** | **값 구분** |
| result | 1 |  | 해당 아이디어에 상태 업데이트에 대한 결과 값 | sucess : 성공(정상)  error : 실패 |
| description | 1 |  | 결과(result 엘리먼트)에 대한 설명 |  |
| code | 1 |  | 종목코드 |  |
| code\_name | 1 |  | 종목명 |  |
| thema\_name | 1 |  | 테마명 |  |
| sub\_price | 1 |  | 가격 |  |
| ROE | 1 |  | ROE |  |
| PER | 1 |  | PER |  |

**\*** 샘플 JSON예제

|  |
| --- |
| - 실패할 경우 샘플  {“result":"error","description":"etc"}  -성공할 경우 샘플  {“result”:”sucess”,”description”:”sucess”,”code”: “1234578”, “code\_name” : “삼성전자”,  “thema\_name” : “반도체”, “sub\_price”:”154525”,“ROE” : “20”, “PER” : “3”  } |

1. 코스닥 종목중 가장 높은 ROE를 가진 종목의 정보를 조회

코스닥 종목중 가장 높은 ROE를 가진 종목의 정보를 조회

|  |  |
| --- | --- |
| URL | /comento/biz/code/maxRoe |
| Content-Type | application/x-www-form-urlencoded; charset=UTF-8 |
| Request Method | GET |

**- Request body**

|  |  |  |  |
| --- | --- | --- | --- |
| **파라미터명** | **타입** | **필수여부** | **설명** |
| code | String | **Y** | 종목코드 |
| market | String | **Y** | market(ex: kospi, kosdaq) |
|  |  |  |  |

* **Response body**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **엘리먼트 명** | **Depth** | **배열**  **구분** | **설명** | **값 구분** |
| result | 1 |  | 해당 아이디어에 상태 업데이트에 대한 결과 값 | sucess : 성공(정상)  error : 실패 |
| description | 1 |  | 결과(result 엘리먼트)에 대한 설명 |  |
| code | 1 |  | 종목코드 |  |
| code\_name | 1 |  | 종목명 |  |
| thema\_name | 1 |  | 테마명 |  |
| sub\_price | 1 |  | 가격 |  |
| ROE | 1 |  | ROE |  |

**\*** 샘플 JSON예제

|  |
| --- |
| - 실패할 경우 샘플  {“result":"error","description":"etc"}  -성공할 경우 샘플  {“result”:”sucess”,”description”:”sucess”,”code”: “1234578”, “code\_name” : “삼성전자”,  “thema\_name” : “반도체”,”sub\_price” : “13500”, “ROE” : “20”} |

1. Appendix 암호화/복호화 방식 참고(java 기준)

**\* 암호화**

|  |
| --- |
| public String encrypt(String plainText, String iv, String key) throws Exception {  SecretKeySpec skeySpec = new SecretKeySpec(key.getBytes(), "AES");  Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");  cipher.init(Cipher.ENCRYPT\_MODE, skeySpec, new IvParameterSpec(iv.getBytes("UTF-8")));  byte[] encryptedText = cipher.doFinal(plainText.getBytes("UTF-8"));  String encryptedStr = new String(Base64.encodeBase64(encryptedText));  return encryptedStr;  } |

* **복호화**

|  |
| --- |
| public String decrypt(String encryptedText, String iv, String key) throws Exception {  SecretKeySpec skeySpec = new SecretKeySpec(key.getBytes(), "AES");  Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");  cipher.init(Cipher.DECRYPT\_MODE, skeySpec, new IvParameterSpec(iv.getBytes("UTF-8")));  byte[] byteEncryptedStr = Base64.decodeBase64(encryptedText.getBytes("UTF-8"));  String plainText = new String(c.doFinal(byteEncryptedStr), "UTF-8");  return plainText;  } |

1. 추가 논의 사항

1. 서버 방화벽 처리(전용선, 인터넷선)

2. 점검 시 처리 방식

3. 초당 API 처리 속도(TPS)

4. 가격(사업에서 진행을 해서 보통은 진행을 안함)

5. 확장성(현재는 이렇게 만들었지만 추후 변경이 될수 있는 내용을 검토)

6. 암호화키 변경 시 전달 방식

7. 한글 인코딩 방식

8. 개발계/UAT 테스트 방법(해당 서버로 연동 요청을 하면 방화벽처리가 된 서버만 처리가

가능해서 테스트를 진행하면 안되는 경우들이 많아서 테스트 방법에 대해서도 논의를 해두면 좋습니다.)

9. 한글 인코딩 방식