**\* 요구사항 :**

1. 전체 투자상품 조회

* 조건 : 상품 모집기간 내의 상품
* 상품ID, 상품제목, 총 모집금액, 현재 모집금액, 투자자 수,

투자모집상태(모집중, 모집완료), 상품 모집기간

1. 투자하기

* 사용자 식별값, 상품ID, 투자금액
* 총 투자모집 금액을 넘어서면 sold-out 상태 응답

1. 나의 투자상품 조회

* 내가 투자한 모든 상품 반환
* 상품ID, 상품 제목, 총 모집금액, 나의투자금액, 투자일시

**\* API 정보 :**

1. 전체 투자상품 조회 ( /api/productlist )

- 요청 : 없음

- 응답 :

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Type | Required | Description |
| productId | Long | O | 상품번호 |
| title | String | O | 상품제목 |
| totalInvestingAmount | Long | O | 총 모집금액 |
| cusInvestingAmount | Long | O | 현재 모집금액 |
| investCnt | int | O | 투자자 수 |
| investStatus | String | O | 투자모집상태 |
| startedAt | LocalDateTime | O | 상품 시작일시 |
| finishedAt | LocalDateTime | O | 상품 종료일시 |
| rangeAt | Long | O | 상품 모집기간 |

- 결과 :

{

    "code": "200",

    "message": "정상 처리",

    "body": [

        {

            "productId": 3,

            "title": "제목3",

            "totalInvestingAmount": 300000,

            "cusInvestingAmount": 0,

            "investCnt": 0,

            "investStatus": "모집중",

            "startedAt": "2021-05-13T08:05:34.479",

            "finishedAt": "2021-05-26T08:05:34.479",

            "rangeAt": 13

        }

    ]

}

- 제약사항 : 상품모집기간내 상품조회하며, 모집기간 이외의 상품은 미노출 확인

2. 투자하기 ( /api/insertInvest )

- 요청 :

|  |  |  |  |
| --- | --- | --- | --- |
| Prameter | Type | Required | Description |
| X-USER-ID | Long | O | 사용자 식별값(Header) |
| productId | Long | O | 상품번호 |
| investAmount | Long | O | 투자금액 |

- 응답 : 투자한 내용 응답 받으며, 총 투자모집 금액을 넘어서면 sold-out 상태 응답

- 결과 1:

{"code": "200",

    "message": "정상 처리",

    "body": {

        "num": 7,

        "userId": 1,

        "productId": {

            "productId": 3,

            "title": "제목3",

            "totalInvestingAmount": 300000,

            "cusInvestingAmount": 20000,

            "investCnt": 1,

            "investStatus": "모집중",

            "startedAt": "2021-05-13T06:40:47.313",

            "finishedAt": "2021-05-26T06:40:47.313"

        },

        "investAmount": 20000,

        "investedAt": "2021-05-16T06:41:06.655"

    }

}

- 결과 2:

{

    "code": "200",

    "message": "정상 처리",

    "body": "SOLD OUT"

}

- 제약사항 : 투자금액이 총 모집금액보다 넘을 경우 SOLD OUT으로 응답 처리

3. 나의 투자상품 조회 ( /api/myinvestlist )

- 요청 :

\* 요청 값 : 사용자 식별값

\* 응답 : 상품ID, 상품 제목, 총 모집금액, 나의투자금액, 투자일시

|  |  |  |  |
| --- | --- | --- | --- |
| Prameter | Type | Required | Description |
| X-USER-ID | Long | O | 사용자 식별값(Header) |

- 응답 :

|  |  |  |  |
| --- | --- | --- | --- |
| Prameter | Type | Required | Description |
| totalInvestingAmount | Long | O | 총 모집금액 |
| title | String | O | 상품제목 |
| productId | Long | O | 상품번호 |
| investAmount | Long | O | 나의투자금액 |
| investedAt | LocalDateTime | O | 투자일시 |

- 결과:

{

    "code": "200",

    "message": "정상 처리",

    "body": [

        {

            "totalInvestingAmount": 100000,

            "title": "제목1",

            "productId": 1,

            "investAmount": 30000,

            "investedAt": "2021-05-16T07:38:09.728"

        }

    ]

}

**\* 데이터 유형 : 정상, 오류 관련 처리를 일관되게 표현되도록 구현**

1. 정상 처리 : ResponseDto.java

- 정상 처리 결과는 body 필드에 담기도록 작성

응답 데이터 :

{

    "code": "200",

    "message": "정상 처리",

    "body": [

        {

            "totalInvestingAmount": 100000,

            "title": "제목1",

            "productId": 1,

            "investAmount": 30000,

            "investedAt": "2021-05-16T07:38:09.728"

        }

    ]

}

2. 에러 처리 : CommonExceptionController.java

- RestControllerAdvice & ExceptionHandler 오류 공통 처리 작성

응답 데이터 :

{

    "errorStatus": "BAD\_REQUEST",

    "errorCode": 400,

    "errorMessage": "필수 헤더값을 확인해주세요."

}

\* 초기 데이터 : InitDB.java

- 상품 정보 등록 및 고객 투자 정보 등록 데이터 생성

\* 테이블 설계 :

1. 상품 정보 테이블 : Product.java

2. 고객 투자 정보 테이블 : UserTranInfo.java

\* 개발 환경 :

OS : WINDOW 10

SPRING BOOT : 2.4.5 Release

JAVA : Version 8

Lombok 설치 필요

DB: H2 Database

- application.properties DB 설정:

spring.datasource.url=jdbc:h2:mem:testdb

spring.datasource.driverClassName=org.h2.Driver

spring.datasource.username=sa

spring.datasource.password=

spring.jpa.database-platform=org.hibernate.dialect.H2Dialect

spring.h2.console.enable=true

\* JUnit TEST 결과 :

