

정산 시스템 설계서

'22.08.12

김동석

1. 개요	2
2. 해결해야 할 문제	3
3. 시스템 목표	3
4. 시스템 특징	3
5. 분석	3
5.1. 업무 정의	3
5.2. AS-IS	5
5.2.1. 마운드 미디어 음원 transaction flow 와 정산	5
5.2.2. 포크라노스 정산 시퀀스	6
5.2.3. 오즈 정산 시퀀스	7
5.2.4. 오즈 정산 유스케이스	9
5.3. 데이터 흐름	9
5.4. TO-BE	9
6. 설계	9
6.1. 공통코드	10
6.2. 네이밍	10
6.3. 공통 자바 패키지 구조	10
6.4. 컨테이너 자바 패키지 구조	10
6.5. 컨테이너 다이어그램	11
6.6. 유스케이스 정의	13
6.6.1. 액터	13
6.6.2. 유스케이스	13
6.7. 도메인 모델	16
6.8. 컴포넌트	18
6.8.1. 백엔드 Back-End	18
6.8.1.1. mybatis CRUD	18
6.8.1.2. 보안	18
6.8.1.3. 시나리오 테스트	19
6.8.2. 파일 매니저	19
6.8.2.1. 파일 저장 정책	19
6.8.2.2. 업로드	19
6.8.2.3. 다운로드	19
6.8.3. 계산기 SDS(Sum-Divide-Subtract???)	20
6.8.4. 메시지 발송기 Message Sender(alpha 범위 아님)	20
6.9. 클래스 설계	20
6.9.1. 공통 🖱️ doxygen 문서 보기	20
6.9.2. 백엔드 🖱️ doxygen 문서 보기	21
6.9.3. 파일 매니저 🖱️ doxygen 문서 보기	21
6.9.3.1. 로더	21
6.9.3.2. 파일 생성	23

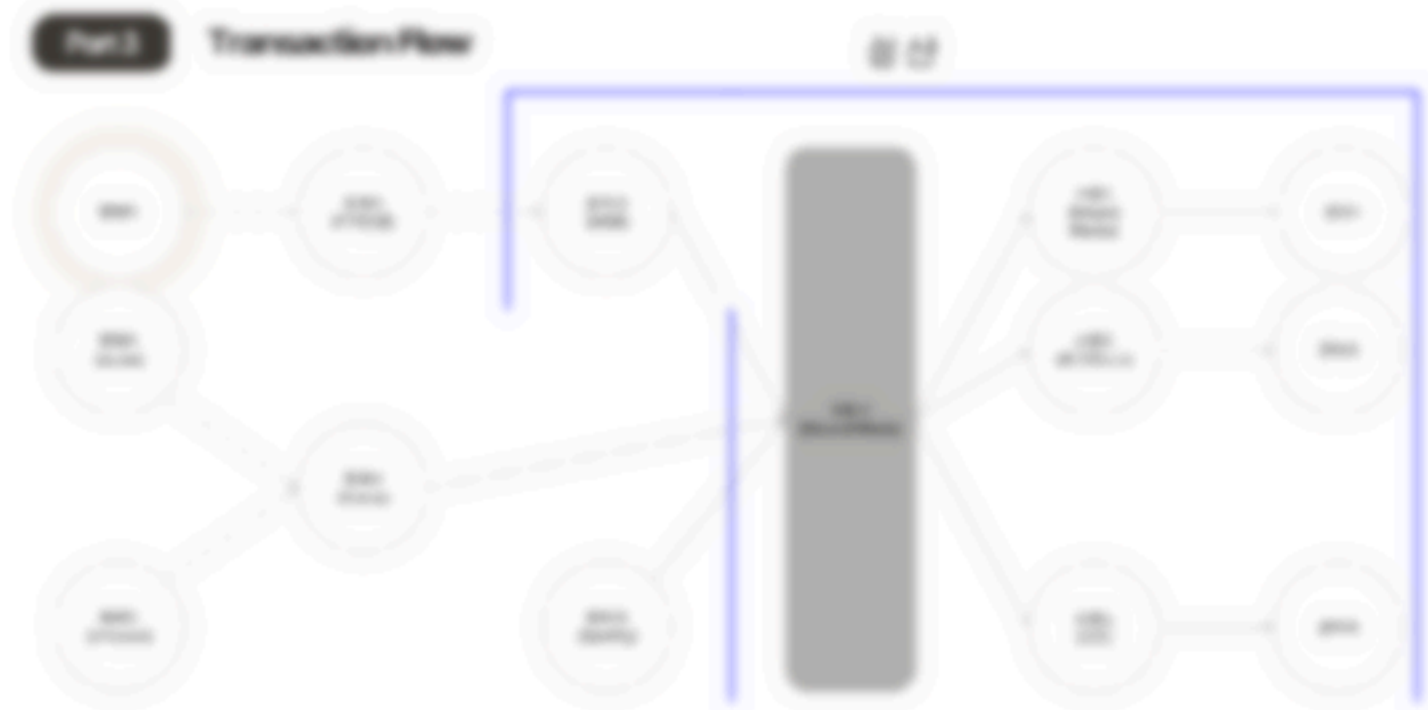
6.9.4. 계산기 SDS(Sum-Divide-Subtract)	24
6.9.4.1. 업무	24
6.10. RESTful API	24
6.10.1. 규약	24
6.10.2. 정산 API 목록	24
6.11. 데이터 유효성 검사	24
6.12. DB	32
6.12.1. fractional seconds rounding problem	32
7. 단어 사전	32
7.1. 금액	33
7.2. 그 외	33
8. 기타	35

6. 2000年10月1日，某公司因出售一项无形资产，取得收入100000元，应交营业税5000元，该无形资产原值为110000元，已摊销60000元。则出售该无形资产应计入营业外收入的金额为（ ）元。

- 가. 동격 관계: 동격, 수식격, 목적격, 주격, 관동격, 관형격, 보충격. 동격의 형태: 주격(主格), 목적격(目的格), 수식격(修飾格), 관동격(關係格), 관형격(關係格), 보충격(補綴格) 설정 가능.
 - 나. 전치격: 전치사(前置詞) 대부분 복수로 전치격 수식명에서 사용 가능.
 1. 수식명 격을 가진
 1. 대부분이 일부 성질(인명명사, 전치 수식명)을 특정 기간으로 격명할 수 있다.
 2. 이성한 대부분이 일대로 동격의 기본 수식명 격명(도움)할 수 있다.
 2. 전치명 대부분 가능하고 동 전치명은 가능하다.
 3. 전치명 수식명에서 지정하는 형식으로 사용 성격이 아닌 다른 형식으로 성격도 가능하다.
- 다. 전치
- 가. 동격: 특정 동격의 수식명 다른 전치사(前置詞)로 지정하는 형식
 - 나. 전치격: 전치격으로 지정 동격의 수식명 동격에서 전치하는 동격
- 물
- 가. 동격: 전치사(前置詞) 수식명의 전치 특정 일부를 전치할 수 있는 경우 다른 동격의 수식명 전치하는 동격
 - 나. 동격
 1. 일대격(一대)으로 일대격 일대
 2. 수식격(修飾)으로 전치사가 사용하는 경우
 3. 일대격
 4. 그 외

5.2. AS-45

5.2.1. DBMS_TRANSACTION SET TRANSACTION Role 10 2021

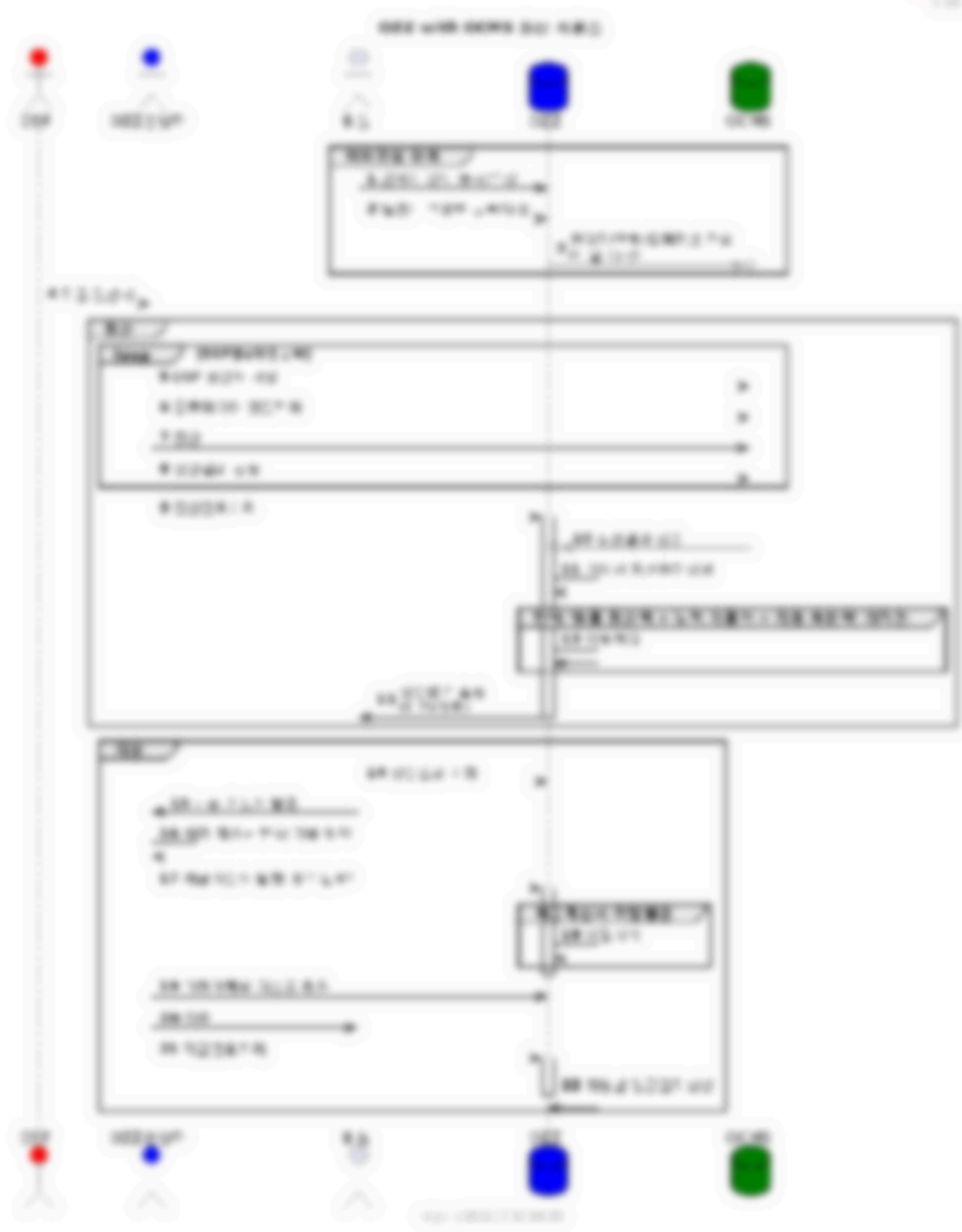


© 2014 Global Transformation, David Newman



5.2.3. 요즈 경산 시원스

Modeling
5.2.3



5.2.4. 모즈 정신 유스 케이스



5.3. 데이터 흐름

1. 사용자: 사용자로부터 수신한 전체 데이터 흐름
2. 사용자: 사용자로부터 수신한 전체 데이터 흐름, 사용자로부터 수신한 데이터 흐름, 사용자로부터 수신한 데이터 흐름
3. 사용자: 사용자로부터 수신한 전체 데이터 흐름, 사용자로부터 수신한 데이터 흐름, 사용자로부터 수신한 데이터 흐름
4. 사용자
5. 사용자: 사용자로부터 수신한 전체 데이터 흐름, 사용자로부터 수신한 데이터 흐름, 사용자로부터 수신한 데이터 흐름
6. 사용자: 사용자로부터 수신한 전체 데이터 흐름, 사용자로부터 수신한 데이터 흐름, 사용자로부터 수신한 데이터 흐름
7. 사용자: 사용자로부터 수신한 전체 데이터 흐름, 사용자로부터 수신한 데이터 흐름, 사용자로부터 수신한 데이터 흐름
8. 사용자: 사용자로부터 수신한 전체 데이터 흐름, 사용자로부터 수신한 데이터 흐름, 사용자로부터 수신한 데이터 흐름
9. 사용자

5.4. TO-BE

6. 설계

* 사용자로부터 수신한 전체 데이터 흐름: user, spring, mybatis, jsp

6.1. 공통 코드

1. ISO3166(국가코드), 4217(통화코드) 같은 국제화된 코드는 거의 변경되지 않는 정보이므로 자바 컨테이너에서 코디 정보를 부하로 저장할 수 있음
 - a. Java: <http://www.iso.org/iso/home/standards/standards.htm>
 - b. Java: <http://www.iso.org/iso/home/standards/standards.htm>
2. 코드 길이 7000

6.2. 네이밍

예: [이름표, 이니셜, 이름, 코드](#) 참조

* 컨테이너 명칭

1. 공통: common
2. 액션드: api
3. 파일 데이터: file
4. 메시지 발송기: msg
5. 로그인: login

6.3. 공통 자바 패키지 구조

Package Name			Description
com.nhn.android.common			기본 package
	conf		공통 정보 관련 package
	controller		Spring MVC controller 관련 package
	domain		도메인 모델 관련 package
	service		비즈니스 로직 관련 package
	mapper		MybatisDB 쿼리 mapping 관련 package
	entity		공통 entity

6.4. 컨테이너 자바 패키지 구조

Package Name			Description
com.nhn.android.container			공통 기본 package
	conf		공통 정보 관련 package
	controller		controller 관련 package
	domain		도메인 모델 관련 package

Warning

경고

이 시스템은 아직 실험 단계입니다

Mound Media 생산시스템 컨테이너



작성 일자 2022-09-26 02:39



- [illegible]

4. 어떤 물질을 열을 가하면 녹아나가는 물질(표준상태에서 부피)
 1. 물
 2. 얼음
 3. 증기
 4. 액체

SYSTEM ARCHITECTURE



6.8. 검토노트

POOD의 자원의 전체이나 일부 작성

6.8.1. 백엔드 Back-End

6.8.1.1. mybatis-CPUED

1. 기존 커넥션풀이었던 것을 변경해 mybatis-generator 사용 [▶ 링크](#)

6.8.1.2. 부가

1. 인증

a. 전제조건

1. Spring Security Form Login 사용
2. 모든 엔드포인트들은 같은 도메인([myunitedmedia.net](#))을 사용하여 부가 공유

b. 인증 방식

1. 부가 back-end의 form-login 주소로 API호출
2. 인증 성공 시 200(OK)로 부가공유 반응
3. 이후의 모든 요청 보디에 이 부가를 인증(도메인이 동일하므로 요청 보디에 명시적으로 부가를 넣지 않아도 인증됨)
4. 예외 처리하면서 다시 form-login API를 호출하여 부가를 받아옴

c. backend

이 인증서 form 로그인 화면으로 리다이렉트(redirect) 되고 나서 받고 받은 데이터를 반환하도록 수정. 또한 받아오는 response body를 반환해야 함

1. ExceptionHandlingConfigurer의 AuthenticationEntryPoint 변경 : SecurityConfig.java.config() 참고
2. AppSecurityConfig의AuthenticationEntryPoint.java : ExceptionHandlingConfigurer에서 사용하는 AuthenticationEntryPoint
3. 참고 [▶ Spring Security 참고](#)

d. frontend

1. API 요청시 부가공유 요청해야 하기 때문에 사용하는 통신객체 Credentials 설정이 있어야 함 [▶ CORS, CSRF, CSRF, CSRF](#)
2. 로그인 요청시 Method : Post, Path : login, Parameter : username, password
3. 로그인 요청시 Method : Post, Path : login
4. backend의 설정에 따라 path, frontend가 이 정보가 달라질 수 있음
5. form-login의 csrf 토큰을 잘 가져올 수 없거나 사용하지 않는다면 backend에서 httpSecurity의 csrf.disable()가 필요 함

2. OAuth

a. OAuth에서 제공하는 OAuth를 통해서 OAuth 인증

b. 제공 domain id

1. [https://oauth2.myunitedmedia.net/](#)
2. [https://oauth2.backend.myunitedmedia.net/](#)

3. 데이터 인증 및

a. 데이터 : 모든 개인정보 정보

- b. 방식 : AES/CBC/PKCS5Padding
- c. 공통 라이브러리 사용

<https://docs.oracle.com/javase/8/docs/api/java/security/spec/BlockCipherSpi.html#AES-CBC-PKCS5Padding-java.lang.String-java.lang.String-java.lang.String->

6.8.1.3. 서니티팅 테스트

- locate 링크 :

<https://docs.oracle.com/javase/8/docs/api/java/security/spec/BlockCipherSpi.html#AES-CBC-PKCS5Padding-java.lang.String-java.lang.String-java.lang.String->

6.8.2. 파일 암호화

6.8.2.1. 파일 저장 정책

TODO 검토 필요

1. 암호화할 데이터 저장?
2. 파일 종류별로 할지 저장?
3. 저장시 이름도 저장 저장으로 변경할지? 파일 확장자는 그대로 둘지

6.8.2.2. 입출력

* 대용량(대용량 저장장치) 처리 필요

1. 블록CBV 읽기 : 파일 이름(file) 처리 필요
2. 입출력 처리
3. 블록 크기 관리, 저장시 블록 크기
4. 대용량 저장장치(분할 저장, 저장)

파일명
파일명

동작 흐름도 : 파일 암호화 - 입출력



6.8.2.3. 디버깅

1. 블록CBV/PKCS5Padding 검증
2. 파일명
3. 입출력
4. 대용량 저장장치

통신시스템 정보공학 : 제1장 정보통신기술



6.8.3. 계산기 SDS(Sum-Divide-Subtract???)

1. 선형적 변환

6.8.4. 메시지-발신기 Message Sender(alpha 버전-4월)

6.9. 클래스 설계

자료 및 관련 문서 : Refer to <http://www.kh.co.kr>

6.9.1. 공통 ★ www.kh.co.kr

1. gWeb 주소 : <http://www.kh.co.kr>
2. 통신 모듈 : com.kh.net.domain
 - a. BaseDomain.java : 모든 데이터 모듈의 공통 클래스(클래스 상속을 받기)
 - b. BaseSender.java : 모든 데이터 모듈의 공통 클래스(클래스 상속을 받기)
 - c. BaseReceiver.java : 모든 데이터 모듈의 공통 클래스(클래스 상속을 받기)
 - d. 데이터 클래스 : BaseDataLoader : com.kh.net.domain.loader
 - i. BaseDataLoader.java : 모든 데이터 모듈의 공통 클래스(클래스 상속을 받기)
 - ii. BaseData.java : 모든 데이터 모듈의 공통 클래스(클래스 상속을 받기)
3. 통신 모듈 모듈 : com.kh.net
 - a. BaseSenderRunner.java : 시스템 모듈의 실행
4. 실행 : net.kh.net.domain
 - a. application.yml
5. 실행 : mybatis
 - a. 모든 CoreFileMapper 모듈의 실행
 - b. 실행
 - i. 모든 모듈의 mapper location 설정의 classpath에 *를 넣기

mybatis:

mapper-locations: **classpath*** "mybatis/mapper/*" xml

6. MybatisCore를 추가

<https://mvnrepository.com/artifact/org.mybatis/mybatis-core>
<https://mvnrepository.com/artifact/org.mybatis/mybatis-spring>

6.9.2. 백엔드

1. gWeb 소스: <https://mvnrepository.com/artifact/org.mybatis/mybatis-core>

6.9.3. 파일 관리자

1. gWeb 소스: <https://mvnrepository.com/artifact/org.mybatis/mybatis-core>
2. jmeter.pdf: <https://mvnrepository.com/artifact/org.mybatis/mybatis-core>

6.9.3.1. 설치

1. 플러그인 다운로드



2. 메인 애플리케이션

- a. `ServletInitializer`: Spring Boot Application
- b. `AppProperties`: 앱 환경 설정
- c. `ConfigurableClassBody`: 플러그인 선언이 이후 필요한 설정
- d. `ServletInitializer`: 주가적인 `ServletInitializer`를 확인하고 실행 여부 확인
- e. `ServletInitializer`: 메인 애플리케이션 실행 로직
- f. `ServletInitializer`: 메인 애플리케이션 실행 로직

g. `SalesDataLoadTestRunner`: 전체 데이터 로드 및 실행.

5. `TestRunner` 실행: 데이터를 `TestRunner` `SalesDataLoadTest`에 전달(데이터 로드).

* 스스 테스트를 실행하기 위해 `reader`가 `loader`를 처리하도록 함(실제 데이터 로드 및 실행).

* 참고: [SQL, Software System, DB, 프로, 데이터, 구조](#).

a. `ExcelReader`

b. `CSVReader`

c. `NewSalesReport`: 데이터 처리 및 분석 결과 보고서(출력).

d. `NewSalesData.java`: 데이터 처리 및 분석 결과 데이터 인터페이스, `ISalesData`가 정의되어 있음.

e. `NewSalesData----`: 데이터 분석 시뮬레이션 및 실행 결과 보고서 `NewSalesDataRunner.java`.

6. `TestRunner` 실행: 데이터를 `TestRunner` `SalesDataLoadTest`에 전달(데이터 로드).

a. `ExcelReader`: 데이터 처리 및 분석 결과 보고서(출력).

b. `CSVReader`: 데이터 처리 및 분석 결과 보고서(출력).

c. `NewSalesReport`: 데이터 처리 및 분석 결과 보고서(출력).

d. `NewSalesData`: 데이터 처리 및 분석 결과 보고서(출력).

7. `TestRunner`

a. `TestRunner`: 데이터 처리 및 분석 결과 보고서(출력).

8. `TestRunner`

* 테스트를 실행하기 위해 [SQL, Software System, DB, 프로, 데이터, 구조](#).

a. `SalesDataLoader`: 데이터 처리 및 분석 결과 보고서(출력).

b. `SalesDataReporter`: 데이터 처리 및 분석 결과 보고서(출력).

c. `--TypeHandler`: 데이터를 데이터에 맞는 형식으로 변환.

6.9.4. 계산기 SDS(Sum-Divide-Subtract)

6.9.4.1. 업무

1. 실행 기준
 - a. QUEUE_CALCULATE_MANAGER
2. 시퀀스 작업
 - a. 전체를 위한 초기 연산 정보 생성
 - b. 전체를 위한 asset, asset key를 수수료(계산 수수료 포함) 정보 생성
 - c. 전체를 위한 asset, asset key를 연산율 정보 생성
3. 종산 금액 계산
 - a. 수수료, 연산율 설정 실행 금액 계산
4. 연산 금액 계산
 - a. 수수료 계산 후 금액을 연산율 기반으로 종산액 연산
5. 시퀀스 작업
 - a. 연산결과 출력 처리

6.10. RESTful API

6.10.1. 규약

1. 종목, 수수료, 시가의 경우 request header의 requesturl에 작업 시의 아이디를 넣어야 함
2. error 처리 : 값으로 code를 사용하시 말고 errorCode를 사용(code는 국제어장용 ID)
 - a. 코드 : <https://github.com/daewoo-finance/daewoo-finance>
 - b.

6.10.2. 종산 API 목록

☞ [API 목록](#) 참조

6.11. 데이터 유효성 검사

☞ 엔지니어들이 모든 장치를 공짜로써 유효성을 유지하면 개발 공수를 줄일 수 있음.

1. backend에서 입력 데이터에 대한 유효성(양적일 여부, 유효성 기준 등...)을 따로 🚩 swagger에 이용하면 client에서 자동 검증됨 ☞ <https://github.com/daewoo-finance/daewoo-finance>

예) 고객 상품 데이터

```
{
  "categoryId": "W_01234",
  "countryCode": "KR",
  "website": "www.gilb.com",
  "age": 12,
  "gender": "M",
```

```

    "facebook": "1-23",
    "emailAddress": "20-23-45-23",
    "image": "800x22-110x40",
    "phones": {
      "111-222-3333",
      "444-555-6666"
    }
  }
}

```

예제 고객 정보 형식

```

{
  "title": "고객",
  "required": {
    "countryCode",
    "phones",
    "lastId"
  },
  "type": "object",
  "properties": {
    "lastId": {
      "title": "일련식번호",
      "maxLength": 20,
      "minLength": 1,
      "type": "string",
      "description": "일련식번호 시작번호 앞"
    },
    "countryCode": {
      "title": "국가코드",
      "maxLength": 3,
      "minLength": 3,
      "type": "string",
      "description": "ISO 3166"
    },
    "email": {
      "title": "이메일",
      "type": "string"
    },
    "age": {
      "title": "나이",
      "maxLength": 20,
      "minLength": 20,
      "type": "integer",
      "format": "int32"
    },
    "gender": {

```



```

    "value": "홍길동",
    "type": "string"
  },
  "feedback": {
    "value": "너는 훌륭해",
    "type": "number",
    "format": "float"
  },
  "description": {
    "value": "한 사람",
    "type": "string",
    "description": "이름 이후 설명",
    "format": "date"
  },
  "age": {
    "value": "후한공백백년",
    "pattern":
    "^(1|2|3-4|5|6|7-8|9-10|11-12|13-14|15-16|17-18|19-20|21-22|23-24|25-26|27-28|29-30|31-32|33-34|35-36|37-38|39-40|41-42|43-44|45-46|47-48|49-50|51-52|53-54|55-56|57-58|59-60|61-62|63-64|65-66|67-68|69-70|71-72|73-74|75-76|77-78|79-80|81-82|83-84|85-86|87-88|89-90|91-92|93-94|95-96|97-98|99-100)$",
    "type": "string"
  },
  "phones": {
    "value": "전화번호",
    "type": "array",
    "description": "목록",
    "items": {
      "value": "전화번호",
      "type": "string",
      "description": "목록"
    }
  }
}
}
}

```

2. backend 쪽으로 보내. DTO를 이용해 JSON 데이터를 생성하고 호출을 받는 부분
3. backend 쪽으로 보내. json-validation으로 검사하고 오류 시 해당 HTTP 상태 코드로 오류 정보 보냄

```

{
  "code": "COMMON_ERROR",
  "status": 100,
  "detail": "Spring Boot Binding 오류",
  "instance": "url=/devpost/customer/test",
  "errors": {
    {
      "code": {

```

```

    "size.customer.countrycode",
    "size.countrycode",
    "size.java.lang.String",
    "size"
  },
  "arguments": {
    {
      "codes": {
        "customer.countrycode",
        "countrycode"
      },
      "arguments": null,
      "defaultMessage": "countrycode",
      "code": "countrycode"
    },
    {
      "code": "size"
    }
  },
  "defaultMessage": "国名を指定してください @size",
  "objectName": "customer",
  "valid": "countrycode",
  "validation": "none",
  "bindingAllowed": false,
  "code": "size"
},
{
  "codes": {
    "size.customer.age",
    "size.age",
    "size.java.lang.Short",
    "size"
  },
  "arguments": {
    {
      "codes": {
        "customer.age",
        "age"
      },
      "arguments": null,
      "defaultMessage": "age",
      "code": "age"
    },
    {
      "code": "size"
    }
  },
  "defaultMessage": "年齢を指定してください @size",
  "objectName": "customer",

```

```

    "valid": "age",
    "rejectedReason": null,
    "bindingAllowed": false,
    "code": "Age"
  },
  {
    "codes": [
      "Mighty.customer.feature",
      "Mighty.feature",
      "Mighty.java.lang.Float",
      "Mighty"
    ],
    "arguments": [
      {
        "codes": [
          "customer.feature",
          "feature"
        ],
        "arguments": null,
        "defaultMessage": "feature",
        "code": "feature"
      }
    ],
    "defaultMessage": "この顧客の年齢値は範囲外(0以上 99以下)です",
    "objectName": "customer",
    "valid": "feature",
    "rejectedReason": 1.234,
    "bindingAllowed": false,
    "code": "Mighty"
  },
  {
    "codes": [
      "assert true.customer.validPhone",
      "assert true.validPhone",
      "assert true.business",
      "assert true"
    ],
    "arguments": [
      {
        "codes": [
          "customer.validPhone",
          "validPhone"
        ],
        "arguments": null,

```



```

    },
    {
      'code': [
        'assert true, customer, valid?endif',
        'assert true, valid?endif',
        'assert true, business',
        'assert true'
      ],
      'arguments': [
        {
          'code': [
            'customer, valid?endif',
            'valid?endif'
          ],
          'arguments': null,
          'defaultMessage': 'valid?endif',
          'code': 'valid?endif'
        }
      ],
      'defaultMessage': '테스트용 유효성 검증의 역할만 수행을 한다. 실제의 경우는  

      *의존성 주입을 위한 유효성 검증*',
      'objectName': 'customer',
      'valid': 'valid?endif',
      'validWhenInvalid': false,
      'bindingWhenInvalid': false,
      'code': 'assert true'
    }
  ],
  {
    'code': [
      'future, customer, expire?endif',
      'future, expire?endif',
      'future, java.time.LocalDate',
      'future'
    ],
    'arguments': [
      {
        'code': [
          'customer, expire?endif',
          'expire?endif'
        ],
        'arguments': null,
        'defaultMessage': 'expire?endif',
        'code': 'expire?endif'
      }
    ],
    'defaultMessage': '테스트용 유효성 검증의 역할만 수행을 한다. 실제의 경우는'
  }
]

```

```

    "objectName": "customer",
    "valid": "invalidate",
    "rejectedValid": "2020-10-01",
    "bindingFailure": false,
    "code": "future"
  }
}

```

4. GET 요청의 유효성 검사(backend → frontend)

예시: 어떤 권한이 있다 유효성 검사 부서가 없이 발생하였을 경우만 유효성

- backend: api의 유효성 검사 api에 대한 번호를 넣음.
- frontend: 유효성 검사를 가져옴
- frontend: api를 통해 유효성 검사 api에 대한 번호를 저장. 이 번호가 어떤 권한에 대한 번호로 유효성 검사 유효성 검사를 다시 가져옴

5. 비즈니스 로직의 유효성 검사

예: 특정 권한을 가진 권한이 있는 권한을 검사

```

{
  "code": {
    "assert true: customer: valid",
    "assert true: valid",
    "assert true: business",
    "assert true"
  },
  "arguments": {
    {
      "code": {
        "customer: valid",
        "valid"
      },
      "arguments": null,
      "default message": "valid",
      "code": "valid"
    }
  },
  "default message": "비즈니스 로직의 유효성 검사, 유효성 검사, 유효성 검사, 유효성 검사, 유효성 검사, 유효성 검사",
  "objectName": "customer",
  "valid": "valid",
  "rejectedValid": false,
  "bindingFailure": false,
  "code": "assert true"
}

```

- code: Assertion
- code: "valid" = 유효성 검사, "valid": "valid"
- default message: 유효성 검사

6. Java 스스 실행

```
//  
+ JavacertTrust를 호출한 결과를 처리 위한 메소드  
+ 반드시 호출 시켜야(그럼에 호출한 결과가 실패)  
+  
getJdk() // Javacert Trust를 호출한 결과  
getJdkIgnore() // Javacert Ignore를 호출한 결과  
JavacertTrust(message="리소스 출처의 적절한 호출을 무시, 남겨놓으면  
* 리소스 출처를 무시하여 실행()")  
public boolean isJdkTrust() {  
    return JdkTrust.startWith(JdkTrust());  
}
```

6. JdkTrust를 실행 시 <https://www.oracle.com/technetwork/java/javase/8-releases-3595860.html>
의 update의 components section에도 올라 있어야 하지만 JdkTrust의 실행 결과가 실패하여 실패 시 실패

7. 링크

- <https://www.oracle.com/technetwork/java/javase/8-releases-3595860.html>
- <https://www.oracle.com/technetwork/java/javase/8-releases-3595860.html>
- <https://www.oracle.com/technetwork/java/javase/8-releases-3595860.html>
- <https://www.oracle.com/technetwork/java/javase/8-releases-3595860.html>

6.12. DB

6.12.1. fractional seconds rounding problem

1. 링크 Java에서 분수값을 소수점까지 처리하는 방법

```
//  
+ LocalDateTime를 사용하여 시간과 날짜를 저장  
+ LocalDateTime -> LocalDateTime -> LocalDateTime -> LocalDateTime  
+ JavacertTrust  
+  
public void setJdkTrust(LocalDateTime endTime) {  
    LocalDateTime -> LocalDateTime -> LocalDateTime -> LocalDateTime  
    this.endTime = LocalDateTime.of(endTime.toLocalDate(), LocalTime.MAX);  
}
```

하지만 LocalDateTime은 시간과 날짜를 저장하는 데만 사용 가능

2. 링크 JdkTrust를 실행 시

```
url: jdbc:mysql:// -> JavacertTrust -> JavacertTrust
```

링크 <https://www.oracle.com/technetwork/java/javase/8-releases-3595860.html>

- 3.

7. 단어 사전

* DDEX 링크

- 단어 검색 : <http://www.ddex.net/02/MemberIndex.html>
- 용어집 : <http://ddex.net/02/MemberIndex.html?termid=1>
- 세부사항 용어집 : <http://www.ddex.net/02/Detail.asp>

7.1. 금액

특정 권리자의 특정 장수들의 ...

단어	설명
판매 금액(Sale Amount)	매 건별 판매액 총합(판매 상품로부터 수집한 장수금액에서 차감)
장수 금액(Pack Amount)	판매액 - 수수료
선금 금액(Prepaid Amount)	권리자에게 지급한 금액, 부록 장수액에서 차감
선금 상회 금액(Prepaid Return Amount)	선금액 총 수지액에서 차감할 금액 = 상회 금액 - 장수 금액
소득 장수금(Fixed Pack Amount)	장수 금액 - 선금 상회 금액
판매 수령액(Revenue Receipt)	이 권리자를 부디 수지 판매받은 금액 = 소득 장수금 - 판매 차감액
판매 차감액(Revenue Payment Amount)	이 권리자에게 수지 판매받은 금액
지급 금액(Payment Amount)	소득 장수금 + 판매 수령액 + 전지 이월 - 차지 이월
차지 이월(Carried-over Amount)	매인 사무를 인해 전 지 달에 지급되지 못한 지급 금액
전지 이월(Forwarded Amount)	매인 사무를 인해 이전 달에 지급되지 못한 지급 금액
실 지급액(Realizer Amount)	총결정 금액(매주 금액, 지급 금액 + 세금) - 매달의 몫
최소 지급액(Minimum Transfer Amount)	이 금액 이하의 지급 금액은 이월한다.
수수료(Fee)	판매액에서 차감할 금액, 판매액에 수수료율을 곱해서 산출, 합치면 매달의 수지액 총
부가가치세(VAT)	Value Added Tax (소득 장수금) x (가치 판매율) x (세율) 의 매달의 몫
소득세(Income Tax)	(소득 장수금) x (가치 판매율) x (세율) x 0.9% 의 매달의 몫
부양세(Residual Tax)	매달 지출액 거주하는 세당주, 사업소를 본 법인 및 주인사업자에게 부과하는 세금 (소득 장수금) x (가치 판매율) x (세율) x 0.9% 의 매달의 몫

7.2. 그 외

단어	설명
IP assets	지적 자산(물리적 자산을 제외한 모든 것)
Account	
Adaptation	원작, 인쇄 시료부터 받은 작품을 인쇄 시료를 바탕으로 4차권을 가진 창작(후물디)을 본의사에 맞게 재창작하는 과정
Asset	상자물, 저작권, 물적(금전, 물건...)
Author	창작자
Brand (POC)	이름과 로고와 내부의 인쇄 조직
Catalog	작품 목록, 목록(인쇄)의 대상이 되는 작품
Common Works Registration (CWR)	a standard format for the registration and revision of musical works
Copyright	저작권
DSP	Digital Service Provider. 인쇄사, 출판 시스템에서는 출판사의 저작권도 포함이 되는 인쇄데이터를 Mount Media에 제공하는 자의 의미임.
Fee	4차권, 작품에 대한 대가를 인쇄사가 흡수하는 수직(rightshold)
Fee rate	4차권율
License	상품 작품에 대한 권리를 주는 행위, 이에 대한 대가로 royalty를 지불함
Licensee	라이선스를 받은 자
Licensor	라이선스를 제공한 자
Master use license	
Mechanical license	
Mechanical royalties	음적이 피본인(저작권)을 해마다 지불
Neighboring rights	저작권과 관련 없을 경우에도 사용되는 권리, 연주자, 가수, 음반 제작사의 권리(이들에 의한 이익을 인정하지 않음)
Originator	창작자
Ownership share	
Performance royalties	음적이 공연 될 때마다 지불

Performing Rights Organizations	저작권과 출판업자를 대신해서 사용자에게 라이선스 수입을 모아주는 단체
pitching song	수익을 목적으로 실제 간접 사용자에게 주는 홍보용어 playing songs for publishers, artists or record label people in hopes of getting them to record or help promote a song recorded
product	음반, 음반 등
recoupment	신곡몰락
revenue	수익
royalty	저작권
rights controllers	저작권 협회, 출판사, 음반사, Music Licensing Companies
rights holder	권리자
royalty (personal) report	출판(보고)서, sales report와는 다른 개념
royalty/licensing fee	cover를 사용한 대가, 일반적으로 사용(판매)당 revenue에 대한 비율로 산정
sales report	판매 데이터, 상품들 판매 물량에 몇 건 팔았는지에 대한 정보, royalty report와는 다른 개념
sync placements	영상물에 삽입된 저작권이 영상물에 sync 맞춰져 있음
writer	작곡가, 싱어, 프로듀서, 편곡사
sync fee	대사(대조, 확인)
Exchange rate	환율
carry over	이월
used money	유용 / 유금

8. 기타

1. [출판권과 저작권](#)