**CQRS 패턴에 의한 데이터 통합**

* 개요 참조
* <http://www.msaschool.io/operation/integration/integration-six/>
* order 서비스(8081)와 delivery 서비스(8082)를 실행한다.

**cd order**

**mvn spring-boot:run**

**cd delivery**

**mvn spring-boot:run**

* orderView 서비스는 모든 이벤트를 수신하여 자신만의 view table 을 구성하는 서비스이다.
* orderView 서비스의 PolicyHandler.java 의 로직을 확인한다.
* 주문을 발송하여 이벤트를 발생시킨다.
  + 주문 발송  
    http localhost:8081/orders productId=1 productName="TV" qty=3
* 카프카에서 이벤트를 확인 한다.

**/usr/local/kafka/bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic shopmall --from-beginning**

* orderView 서비스를 실행한다.

**cd orderView**

**mvn spring-boot:run**

* orderView의 Query Model을 통해 주문상태와 배송상태를 통합 조회한다.

**http localhost:8090/orderStatuses**

**Compensation Trx 발행**

* 주문 취소

**http DELETE localhost:8081/orders/1**

* 주문상태와 배송상태 값을 확인

**http localhost:8090/orderStatuses**

**Compensation Trx을 Query모델에 반영**

* OrderView에 Compensation Logic에 대해서도 Tracking 되도록 코드를 보완 하세요.
* OrderView의 PolicyHandler.java 수정

**Service Clear**

* 다음 Lab을 위해 기동된 모든 서비스 종료
* 8090 및 808x의 모든 Process Kill

**fuser -k 8090/tcp**

**kill -9 `netstat -lntp|grep 808|awk '{ print $7 }'|grep -o '[0-9]\*'`**

Checkpoints

1. OrderCancelled 이벤트, DeliveryCancelled 이벤트가 왔을때 orderStatus 를 'OrderCancelled','DeliveryCancelled' 로 만드는 로직을 작성 후, 주문을 취소하여 상태값을 변경하시오.