REST는 리소스, 메서드(CRUD 해당post, delete, get, update), 메시지의 3가지 요소로 구성

* HTTP 호출 예제 : “이름이 apexacme인 User를 생성한다.”
* REST 표현  
  "HTTP POST, http://my-web/users/  
  {  
  “users”:{  
  “name”:“apexacme”  
  }  
  } "

httpie를 사용한 HTTP 메서드 예시

* 생성: POST  
  "http POST, <http://myweb/users/>  
  {    
     “id”:“apexacme”,  
  “name”: “Park Yong Joo”,  
     “address”:“seoul”  
  }"
* 조회: GET  
  "http GET, <http://myweb/users/apexacme>"
* 수정: PUT(리소스의 전체 교체), PATCH(자원의 부분 교체)
  + 일부만 바꿀때는 patch (putd로 하면 나멎는 초기화됨)  
    “http PUT, <http://myweb/users/apexacme>  
    {       
       “address”:“pusan”  
    }”
* DELETE: 삭제  
  "http DELETE, <http://myweb/users/apexacme>"

http get http://jsonplaceholder.typicode.com/todos

http post http://jsonplaceholder.typicode.com/todos title="test" userID = "10 completed="true"

**도구(MSA-Ez)를 활용한 CNA 개발**

tool에서 이벤트 스토밍- > 코딩

참고영상 시청

Policy : 주문 발생시 연쇄반응. 타팀에 전달되는…

커멘드 : 이벤트 유발하는 유저 디시젼

Aggregation : 함수

컨텍스트 맵핑 (바운디드 컨텍스트 간의 릴레이션 잡아주는 것 )

라일락(인풋이 됨) 주체가 되는 팀이 라일락을 가져옴.

인터페이스 방식 : req/res(동기), pub/sub(비동기)

비동기 처리시 kafka(메시지 큐)에 던져놓으면… 나중에 가져가서 처리함.

Msaez.io에서 모델링 하고, 소스 다운로드 받아서, 올리면… 합격?

아마존 관리콘솔 접속

Ubuntu 18 버전 실행

서비스 iam

왼쪽메뉴에[서 사용자 클릭

본인 계정 클릭

액세스 키 만들기… 비밀 액세스 키는 팝업 띄우면 나옴.

이벤트 스토밍

조회됨은 view.. CRUD에서 R을 제외한 걸 이벤트로 정리함

이벤트 : 우리팀이 발행

팔러시 ; 타 팀에 관심이 있는..

커멘드는 이벤트를 거의 따라감

액터

어그리게이트 : dB , 엔티티 라고 보면 됨 ( 블랙박스 ) -> 모델링을 해주어야 함

보더랑 엔터티를 합쳐줘요

어그리게이트 모델링 해주어야 함

이벤트 나갈떄 같이 나가야 할 애를 정의해야해

Post persist할 때에 id, name, stock이 kafka가 productchenage가 퍼브리시 될떄, 데이터 가져ㄱ다가 쓰는거

객체가 직렬화되어서 kafka로 나감

코드를 리뷰해보면, java 소스가 생성이 되었지

일반적인 cna 구현

<https://start.spring.io/>

**Dependencies**A 아래 3개 추가

* **Spring Data JPA SQL**Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.
* **H2 Database SQL**Provides a fast in-memory database that supports JDBC API and R2DBC access, with a small (2mb) footprint. Supports embedded and server modes as well as a browser based console application.
* **Rest Repositories WEB**Exposing Spring Data repositories o

Generate

IntelliJ에서 다운받은거풀어서, 프로젝트 오픈

먼저, Aggregate 를 생성 합니다.

http get <http://localhost:8080/products>

http post <http://localhost:8080/products> name=”TV”, stock =10

* + 1번이 만들어져요.

http get <http://localhost:8080/products>/1

http delete <http://localhost:8080/products>/1

이벤트 만들때도 자바 클래스 만들고 소스 붙여넣고, generate 한다.

Postpersist 추가하고,

Kafka(백본에 준하는 인프라 미들웨어, ex. ESB에 준하는… 최근 안정성, 빅데이터에 최적화되어 있어서 많이 써)를 설치해보자

이벤트 기반 메시지 채널 - 카프카(Kafka)

설치

주키퍼 실행

디펜던시, 디펜던시매니저, Hoxton 추가

Application.properties를 applicaotn.yml로 바꿔서 카프카 관련 설정을 저장해준다.

인, 아웃 채널으 잡아주는거야

동일한 topic으로 (shop) 잡고

그룹은 카프카에 채너을 꽂을 때,

그룹이 동일하면 한놈에게만 열어준다

리플리카가 10개일 때, 열개가 다 메시지를 수용하면 열번 처리가 되겠지 -> 그룹당 한놈한테만 이벤트를 열어줍니다.

[kafka 설치폴더]/bin/windows/kafka-console-consumer --bootstrap-server localhost:9092 --topic shop --from-beginning

* + Kafka에 쌓이는 메시지?를 확인할 수 있어요…

주키퍼,

카프카

모니터

**@StreamListener(Processor.INPUT)**

**public void** onEventByString**(@Payload String productChanged){**

**System.out.println(productChanged);**

**}**

**Policy?**

* + 리슨할 인풋 채널을 모니터하는거야 .

코드리뷰를 하면…

게이트웨이…가 노아…

게잍으웨이가 마이크로 서비스 내부로 이넡페이스 처리해준다.

소스 팡리 다운로드

마이크로서비스별로 랭귀지를 선택할 수 있어요.

스프링 부트로 하면 간견ㅀ게 코드 가져갈수 있음

깃은 노드 js로 만들거나 할 때 다양하게 만들어서 쓸수 있어요

CQRS : 마이페이지, 대쉬보드 같은거… 배송팀, 주문팀으로 따로 있는 정보를 가져와야 해~ CQRS로..