**Composite 서비스에 의한 데이터 통합**

* **개요 참조**
  + [**http://www.msaschool.io/operation/integration/integration-five/**](http://www.msaschool.io/operation/integration/integration-five/)
* **주문,상품,배송 서비스를 모두 기동한다.**
  + **터미널 3개를 열어서 각각의 프로젝트로 이동한 후, run을 실행한다.**
  + **주문서비스 기동(8081)**

**cd reqres\_orders**

**mvn spring-boot:run**

* **상품서비스 기동(8085)**

**cd reqres\_products**

**mvn spring-boot:run**

* **배송서비스 기동(8082)**

**cd reqres\_delivery**

**mvn spring-boot:run**

* **1개의 주문을 생성한다.**

**http localhost:8081/orders productId=1 quantity=1 customerId="1@uengine.org"**

* **composite\_service 서비스를 기동(8088) 한다.**

**cd composite\_service**

**mvn spring-boot:run**

* **CompositeService.java 파일의 getOrderByCustomerId 메서드 내용을 파악한다.**
* **getOrderByCustomerId 메서드안에서 3개의 서비스를 모두 호출하고, 데이터를 수집하여 보여준다.**

**http localhost:8088/composite/orders/1@uengine.org**

**장애전파 확인**

1. **주문,배송,상품중 1개의 서비스라도 동작을 안하게 되면 에러가 발생한다.**
2. **3개중 1개의 서비스에서 호출이 늦어지면 전체 조회는 늦어진다.**

* **reqres\_products 서비스의 ProductController.java 파일의 productStockCheck 을 확인 한다.**
* **thread.sleep 부분을 주석 해제 한 후, reqres\_products 서비스를 재시작한다.**
* **한개의 서비스가 0.5초 느려졌지만 컴포지트 서비스를 통하여 데이터를 가져오는 api도 전파된 시간만큼 느려진다.**

**단점 경험 정리**

* **주문, 배송, 상품 서비스가 모두 가동중이어야 데이터 조회가 된다.**
* **주문이력이 많을시에 모든 데이터를 조회 하기때문에 시간이 많이 걸린다.**
* **각 호출 API 별로 return 되는 data 를 알고 있어야 한다. ( 각 서비스에서 변경시 잦은 변경 요청)**
* **1개의 서비스에서 호출이 늦어지면 전체 조회는 늦어진다.**

**Service Clear**

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

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

**구현관련 시사점**

1. **HATEOAS 링크: 각 주문건에 대한 배송정보를 links.delivery.href 로 제공해줄 수 있도록 하기 위한 구현:**

* **order의 config/Config.java**

**@Bean**

**public ResourceProcessor<Resource<Order>> orderProcessor() {**

**return new ResourceProcessor<Resource<Order>>() {**

**@Override**

**public Resource<Order> process(Resource<Order> resource) {**

**resource.add(new Link("/deliveries/search/findByOrderIdOrderByDeliveryIdDesc?orderId=" + resource.getContent().getId(), "delivery"));**

**return resource;**

**}**

**};**

**}**

**참고: Graph QL in Java Spring**

[**https://www.baeldung.com/spring-graphql**](https://www.baeldung.com/spring-graphql)

**CheckPoints**

1. 모든 서비스를 실행하고 http localhost:8088/composite/orders/1@uengine.org 를 호출한다.

2. 배송서비스를 중지하고 1번 채크항목을 실행하여, 장애전파를 확인한다.