* 폴리글랏 퍼시스턴스

팀프로젝트 진행 시 간편한 DB구성을 위해 RDBMS 기반의 H2 DB를 적용하였다. H2는 Docker에서 설정이 가능하기 때문에 application.yml 파일에는 설정하지 않았으며 dependencies에만 추가하여 진행하였다.

@Table를 사용하여 따로 테이블명을 지정하였으며 Entity Pattern과 Repository Pattern을 적용하였다.

|  |
| --- |
| **Product.java** |
| package rentalService;  @Data  *@Entity*  *@Table*(name="Product\_table")  public class Product {  *@Id*  *@GeneratedValue*(strategy=*GenerationType*.***AUTO***)  private Long id;  private String name;  *@ColumnDefault*("10") //default 10  private int qty ;  } |

|  |
| --- |
| **ProductRepository.java** |
| package rentalService;  import org.springframework.data.repository.PagingAndSortingRepository;  public interface ProductRepository extends PagingAndSortingRepository<Product, Long>{  } |

|  |
| --- |
| **pom.xml** |
| <dependency>  <groupId>com.h2database</groupId>  <artifactId>h2</artifactId>  <scope>runtime</scope>  </dependency> |

* 폴리글랏 프로그래밍

물품 대여 시스템의 시나리오인 대여, 배송 등의 시스템 구현 방식은 JPA를 기반으로 구현하였으며 주요 이벤트 처리방식은 Kafka, FeignClient를 적용하였다.

[Kafka 적용]

|  |
| --- |
| **kafkaProcessor.java** |
| package rentalService.config.kafka;  import org.springframework.cloud.stream.annotation.Input;  import org.springframework.cloud.stream.annotation.Output;  import org.springframework.messaging.MessageChannel;  import org.springframework.messaging.SubscribableChannel;  public interface KafkaProcessor {  String INPUT = "event-in";  String OUTPUT = "event-out";  @Input(INPUT)  SubscribableChannel inboundTopic();  @Output(OUTPUT)  MessageChannel outboundTopic();  } |

|  |
| --- |
| Rental.java |
| @PostPersist  public void onPostPersist(){  Rentaled rentaled = new Rentaled();  BeanUtils.copyProperties(this, rentaled);  rentaled.publishAfterCommit();  } |

[FeingClient 적용]

|  |
| --- |
| DeliveryService.java |
| package rentalService.external;  import org.springframework.cloud.openfeign.FeignClient;  import org.springframework.web.bind.annotation.RequestBody;  import org.springframework.web.bind.annotation.RequestMapping;  import org.springframework.web.bind.annotation.RequestMethod;  import java.util.Date;  @FeignClient(name="Delivery", url="${api.delivery.url}")  //@FeignClient(name="Delivery", url="http://localhost:8083")  public interface DeliveryService {  @RequestMapping(method= RequestMethod.POST, path="/deliveries")  public void deliveryCancel(@RequestBody Delivery delivery);  } |