Date : 28-May-21
Spring Boot 9AM
Mr. RAGHU

Spring Boot + Apache Kafka

1. KafkaTemplate<K,V> :-
 This class is used to send data in key=val format which creates internally
 ProducerRecord<K,V> and send to Kafka S/w.

 K=TopicName, V= Data which is Serialized and trasfred.

2. @KafkaListener (topicName) :
 It reads data from Kafka S/w thatc comes in ConsumerRecord<K,V>.
 Data given Deserialized format.

*) groupId is used for multiple consumers to have data replication.
  We can provide replication factor while creating topicName, else
  use groupId.

Dependencies: DevTools, Lombok, MySQL, Data JPA
                Web, Spring for Apache Kafka,
--coding files order--------------------
1. StockQuote Model

```
package in.nareshit.raghu.model;

import java.util.Date;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;

import lombok.Data;

@Entity
@Table(name="stock_quote_tab")
@Data
public class StockQuote {
        @Id
        @GeneratedValue(strategy = GenerationType.IDENTITY)
        private Integer id;

        @Column(name="stk_code_col")
        private String stockCode;

        @Column(name="stk_cost_col")
        private Double shareValue;

        @Column(name="stk_dte_col")
```

```
        @Temporal(TemporalType.TIMESTAMP)
        private Date  serviceDate;

}
```

2. StockQuote Repository

```
package in.nareshit.raghu.repo;

import org.springframework.data.jpa.repository.JpaRepository;

import in.nareshit.raghu.model.StockQuote;

public interface StockQuoteRepository
        extends JpaRepository<StockQuote, Integer>{

}
```

3. MessageStoreService

```
package in.nareshit.raghu.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import in.nareshit.raghu.model.StockQuote;
import in.nareshit.raghu.repo.StockQuoteRepository;

@Service
public class MessageStoreService {

        @Autowired
        private StockQuoteRepository repository;

        public void addStockData(StockQuote sq) {
                repository.save(sq);
        }

        public List<StockQuote> getAllStockQuotes() {
                return repository.findAll();
        }
}
```

4. JsonUtil

```
package in.nareshit.raghu.util;

import org.springframework.stereotype.Component;

import com.fasterxml.jackson.core.JsonProcessingException;
import com.fasterxml.jackson.databind.ObjectMapper;

import in.nareshit.raghu.model.StockQuote;
```

```
@Component
public class JsonUtil {

	public String toJson(StockQuote sq) {
		try {
			return new
ObjectMapper().writeValueAsString(sq);
		} catch (JsonProcessingException e) {
			e.printStackTrace();
		}
		return null;
	}

	public StockQuote fromJson(String json) {
		try {
			return new ObjectMapper().readValue(json,
StockQuote.class);
		} catch (Exception e) {
			e.printStackTrace();
		}
		return null;
	}
}
```

5. ProducerService

```
package in.nareshit.raghu.producer;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.kafka.core.KafkaTemplate;
import org.springframework.stereotype.Component;

import in.nareshit.raghu.model.StockQuote;
import in.nareshit.raghu.util.JsonUtil;

@Component
public class ProducerService {

	private static final Logger LOG =
LoggerFactory.getLogger(ProducerService.class);

	@Value("${my.app.tpcName}")
	private String topic;

	@Autowired
	private KafkaTemplate<String, String> template;

	@Autowired
	private JsonUtil util;

	public void sendData(StockQuote sq) {
```

```
                String message = util.toJson(sq);
                LOG.info("AT PRODUCER RECEIVED {}",message);
                template.send(topic, message);
        }
}
```

6. ConsumerService

```
package in.nareshit.raghu.consumer;

import java.util.Date;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.kafka.annotation.KafkaListener;
import org.springframework.stereotype.Component;

import in.nareshit.raghu.model.StockQuote;
import in.nareshit.raghu.service.MessageStoreService;
import in.nareshit.raghu.util.JsonUtil;

@Component
public class ConsumerService {

        private static final Logger LOG =
LoggerFactory.getLogger(ConsumerService.class);

        @Autowired
        private MessageStoreService service;

        @Autowired
        private JsonUtil util;

        @KafkaListener(topics = "${my.app.tpcName}",groupId =
"groupId")
        public void readData(String message) {
                LOG.info("DATA AT CONSUMER {}",message);
                StockQuote sq =  util.fromJson(message);
                sq.setServiceDate(new Date());
                service.addStockData(sq);
        }
}
```

7. RestController

```
package in.nareshit.raghu.rest;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import in.nareshit.raghu.model.StockQuote;
import in.nareshit.raghu.producer.ProducerService;
import in.nareshit.raghu.service.MessageStoreService;

@RestController
@RequestMapping("/api/quote")
public class StockQuoteRestController {

	@Autowired
	private ProducerService producer;

	@Autowired
	private MessageStoreService storeService;

	//1. send StockQuote
	@PostMapping("/create")
	public ResponseEntity<String> createStockQuote(
			@RequestBody StockQuote stockQuote)
	{
		producer.sendData(stockQuote);
		return ResponseEntity.ok("Quote Data is sent!");
	}

	//2. view all received
	@GetMapping("/all")
	public ResponseEntity<List<StockQuote>> getAllQoutes() {
		List<StockQuote> list =
storeService.getAllStockQuotes();
		return ResponseEntity.ok(list);
	}

}
```

8. proeprties file

```
#Server
server.port=8761
# Database
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/boot9am
spring.datasource.username=root
spring.datasource.password=root

# JPA
spring.jpa.show-sql=true
spring.jpa.hibernate.ddl-auto=update
spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect
# Kafka (Producer , Consumer)
spring.kafka.producer.bootstrap-servers=localhost:9092
spring.kafka.producer.key-
serializer=org.apache.kafka.common.serialization.StringSerializer
spring.kafka.producer.value-
```

```
serializer=org.apache.kafka.common.serialization.StringSerializer

spring.kafka.consumer.bootstrap-servers=localhost:9092
spring.kafka.consumer.key-
deserializer=org.apache.kafka.common.serialization.StringDeserializer
spring.kafka.consumer.value-
deserializer=org.apache.kafka.common.serialization.StringDeserializer

# topic name
my.app.tpcName=sample-tpc-nit

------------Execution Steps------------------------------
1. Start Zookeeper
2. Start Kafka server
3. Run your application
4. Send data using POSTMAN
---------------------------------------------
POST  http://localhost:8761/api/quote/create  SEND
     Body
            raw(*)  [JSON]

{ "stockCode" : "NIT-INFRA LTD", "shareValue" : 1524.50 }

---------------------------------------------
GET  http://localhost:8761/api/quote/all  SEND

5. check in db table
```