

Apache Camel - Saga sample

Springboot + Saga Development Test

Camel Framework

아파치 카멜(**Apache Camel**)은 라우팅, 중재 규칙 구성을 위한 **API**를 사용하여 **Enterprise Integration Patterns**의 자바 객체 기반 구현체를 제공하는, 규칙 기반 라우팅 및 중재 엔진을 갖춘 메시지 지향 미들웨어를 위한 오픈 소스 프레임워크이다. - [wiki](#)

From

File
Active MQ
Rest API
Kafka
...

Exchange

Header
http..
Body
String, Object
Json, Xml

To

File
Active MQ
Rest API
Kafka
...

컴포넌트설정

Maven Pom.xml 안에 필요한 Component들만 Dependency 추가하여 사용 가능

```
<properties>
  <maven.compiler.source>11</maven.compiler.source>
  <maven.compiler.target>11</maven.compiler.target>
  <spring-boot.version>2.3.4.RELEASE</spring-boot.version>
  <camel.version>2.23.2</camel.version>
</properties>

<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-dependencies</artifactId>
      <version>${spring-boot.version}</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>
    <dependency>
      <groupId>org.apache.camel</groupId>
      <artifactId>camel-spring-boot-dependencies</artifactId>
      <version>${camel.version}</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>
  </dependencies>
</dependencyManagement>
```

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
  <exclusions>
    <exclusion>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-tomcat</artifactId>
    </exclusion>
  </exclusions>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-undertow</artifactId>
</dependency>
<!-- Camel -->
<dependency>
  <groupId>org.apache.camel</groupId>
  <artifactId>camel-lra-starter</artifactId>
</dependency>
<dependency>
  <groupId>org.apache.camel</groupId>
  <artifactId>camel-jackson-starter</artifactId>
</dependency>
<dependency>
  <groupId>org.apache.camel</groupId>
```

RestAPI 사용 시 Camel vs SpringBoot 진입점 비교

```
@Override
public void configure() throws Exception {
    restConfiguration()
        .apiContextPath("/api-doc")
        .apiProperty("api.title", "PURCHASES REST API")
        .apiProperty("api.version", "1.0")
        .apiProperty("cors", "true")
        .apiProperty("base.path", "camel/")
        .apiProperty("api.path", "/")
        .apiProperty("host", "")
        .apiContextRouteId("doc-api")
        .component("componentId: "servlet")
        .bindingMode(RestBindingMode.json)
        .dataFormatProperty("prettyPrint", "true");

    rest(path: "/purchases").description(text: "Purchases endpoint") RestDefinition
        .produces(mediaType: "application/json")
        .post(uri:("/{itemId}/{quantity}") .outType(Order.class).description(text: "Request a new purchase")
            .param().name("itemId").description(name: "Item identification").type(RestParamType.path)
            .param().name("quantity").description(name: "Item quantity").type(RestParamType.path)
            .route().routeId("purchaseRoute") RouteDefinition
            .to(uri: "direct:buy");

    from(uri: "direct:buy") RouteDefinition
        .routeId("buyRoute")
        .log(">>> ${body}")
}
```

Camel - RouterBuilder

```
@PostMapping("/deposit-offer")
public ResponseEntity<DepositOfferResponseSchema> depositOffer (
    @RequestBody DepositOfferRequestSchema depositOfferRequestSchema
){
    DepositOfferServiceRequestDto req = DepositOfferServiceRequestDto.builder()
        .customerId(depositOfferRequestSchema.getCustomerId())
        .productCode(depositOfferRequestSchema.getProductCode())
        .build();

    HttpServletRequest servletRequest = ((ServletRequestAttributes) RequestContext
        .getRequest()).getServletRequest();
    log.info("request : {}", servletRequest);
    log.info("Thread Information -> {}", Thread.currentThread());

    RequestAttributes requestAttributes = RequestContextHolder.getRequestAttributes();

    // Saga Route 등록
    DepositOfferServiceResponseDto res =
        template.withHeader("customerId", depositOfferRequestSchema.getCustomerId())
            .to(endpointUri: "direct:controllerTest")
            .withBody(req)
            .request(DepositOfferServiceResponseDto.class)
            .getBody();
    log.info("Done");
}
```

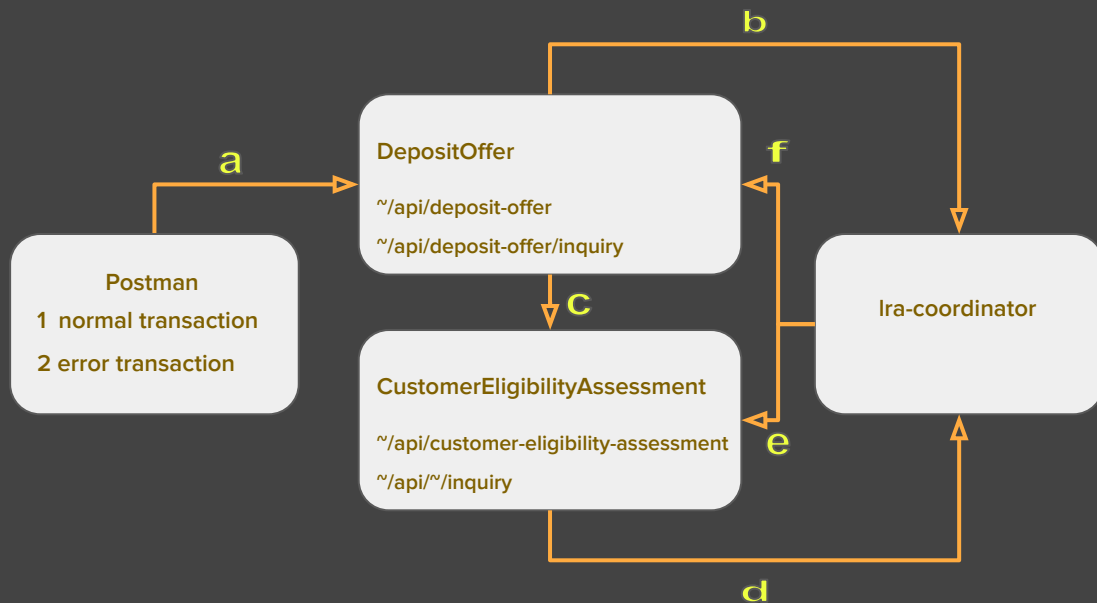
Spring - Controller

사용 Tool & 참고 Reference

- Postman (Rest API Call)
- IntelliJ (2 microservice Run)
 - Spring-Boot Application Run (with Camel)
 - <https://camel.apache.org/>
 - H2 database in-memory (Store data only during runtime)
- Docker (Ira-coordinator)
 - <https://www.narayana.io/Ira/>
 - <https://hub.docker.com/r/jbosstm/Ira-coordinator>
- Github (sample source repository)
 - <https://github.com/Hosuk-Lee/saga-Ira-springboot/tree/feature/spring-camel>
 - <https://github.com/jangminjun/saga-Ira-demo>
 - <https://github.com/jangminjun/saga-Ira-spring-test>

SpringBoot with Camel Saga 샘플 프로젝트

2개의 microservice (DepositOffer, CustomerEligibilityAssessment) 와 1개의 Ira-coordinator 사용



거래 흐름

- 정상거래 (신규생성) 오류거래 (고객등급 오류) 호출
- DepositOffer가 신규 처리요청 수신
camel-saga를 통해 Ira-coordinator transaction 참가신청
camel-header 부에 Long-Running-Action 수신
- CustomerEligibilityAssessment로 처리요청
b에서 수신된 Ira key값을 http header로 전파
- CustomerEligibilityAssessment가 신규 처리요청 수신
Ira 값이 있으면 transaction에 상속되어 참가
- 정상거래시 complete 호출
보상거래시 compensate 호출
- 상동

추가사항

- Camel-Saga d,e 단계에서 정상 또는 오류보고를 하게됨.
- 오류 발생 시 Stack 처럼 역순으로 보상 거래 수행
- 모든 거래가 완료되어야 Ira가 close 되어 사라짐
- inquiry로 데이터 조회

DepositOffer		
id	table key	
createdAt	생성일	
modifiedAt	수정일	
customerId	고객ID (input)	
productCode	상품코드 (input)	
openDate	개설일 (Today)	
customerEligibilityAssessment	서비스호출	
status	응답상태	
message	응답메시지	
customerGrade	고객등급	
status	원장상태	
message	메시지	
lrald	LRA 키값	동일 Key Check

CustomerEligibilityAssessment		
id	table key	
createdAt	생성일	
modifiedAt	수정일	
customerId	고객ID (input)	100010001 100010002 100010003 100010004 오류 100010005 오류
productCode	상품코드 (input)	06 고정
assessmentAt	평가일 (Today)	
status	원장상태	
message	메시지	
lrald	LRA 키값	동일 Key Check

문제점 발견 2개

- **Bugs** : Service 첫 Request 시 멈춤 현상 발생
- **Critical** : Tomcat(undertow)에서 RequestServletContext가 생성되어 생성주기동안 메모리공유가 되어 SpringFramework가 사용 또는 비지니스에서도 사용가능한데 Camel 에 Saga 등록 시 새로운 Thread가 생성되어 사용 불가.
원인 : Java Thread Local 참고

미해결 사항

- 오류처리 (Camel Exception을 Spring 으로 전달 하는 법)
-> Camel 문법 숙지 후 오류처리 개발
- Spring Context (Security, Request) 사용불가로 Spring Reference 참조 불가
-> Critical 이슈 해결 후 조치 방안 도출

잠재이슈

- LRA 모니터링 방법 및 클러스터링 구성 여부 (lra-coordinator 엔지니어 지원 가능여부)
- 배포 버전이 낮음. (Redhat Fuse 가이드 프로그램 SpringBoot 2.3.4 (2.7.4) , Camel 2.23.2 (3.x))



THANK YOU

Hosuk.Lee hosuk@kbfg.com