IBM webMethods Kafka Adapter 구성 및 연결

> **References**
>
> - **Kafka Adapter Documentation**
> - **Kafka Replication Factor (참고)**
> - **JAAS (Java Authentication and Authorization Service)**
> - **Kafka SASL 프레임워크**
> - **Kafka Auto Commit**
> - **Kafka Auto Offset Reset**

※ Kafka Adapter 설치 라이브러리

→ 반드시 설치 시 Fix 필요

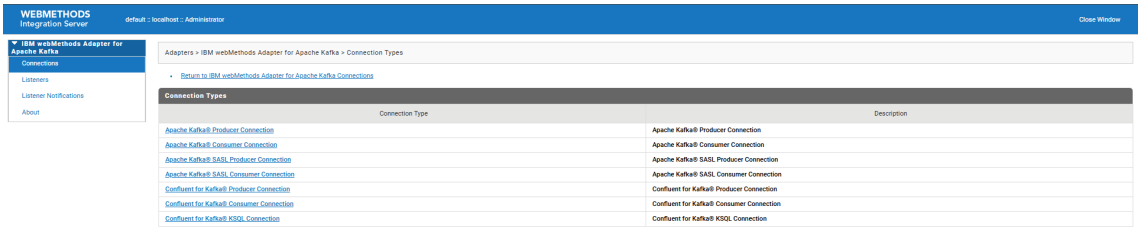
→ 설치 된 Kafka 에서 libs 폴더 아래 버전에 맞는 라이브러리를 복사 or 담당자 에게 요청

- kafka-tools-3.x.x.jar
- kafka-clients-3.x.x.jar
- kafka_2.12-3.x.x.jar

{Integration Server_directory}/instances/default/packages/WmKafkaAdapter/code/jars 아래 해당 라이브러리 업로드 후 재기동

❖ Kafka Adapter Connection

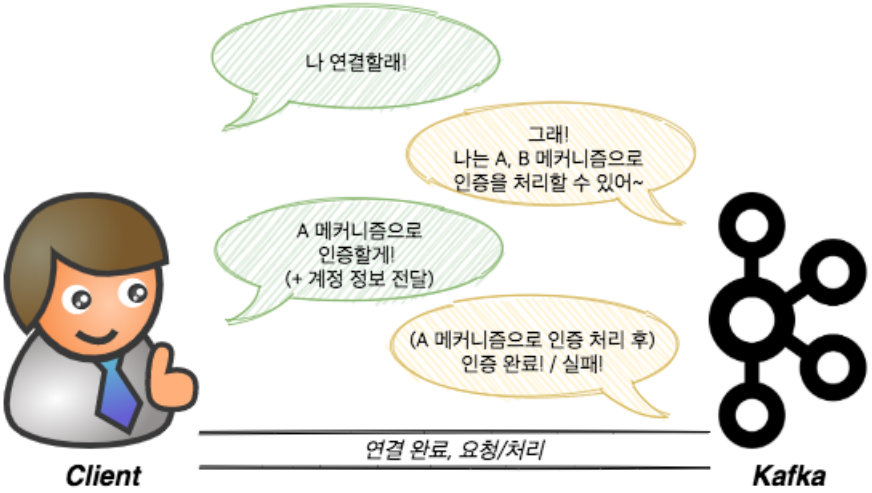
(IBM webMethods Adapter 9.6 for Apache Kafka Fix 16 기준)



webMethods Kafka Adapter 는 4가지 방식의 Connection 을 제공한다.

Apache Kafka Connection (Producer/Consumer)	Apache Kafka 기본 연결 방식 (보안 X)
Apache Kafka SASL Connection (Producer/Consumer)	**Apache Kafka SASL 프레임워크 연결 방식**
Confluent for Kafka Connection (Producer/Consumer)	Confluent 플랫폼 연결 (Confluent 는 Apache Kafka 기반 이벤트 스트리밍 플랫폼 솔루션)
Confluent for Kafka KSQL Connection	Confluent 에서 사용하는 KSQL 방식 연결 (KsqlDB, 유사 쿼리문, Push, Pull Query 로 구성)

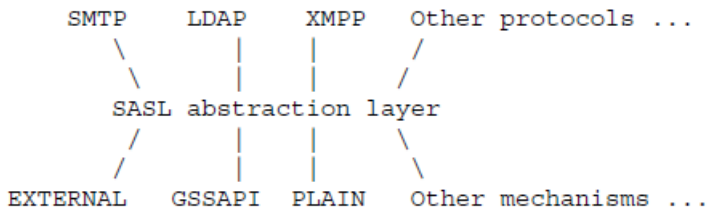
Note 1. Kafka 의 인증 방식의 이해 (SSL/SASL)



Client = webMethods

- SSL - 인증서를 이용한 인증 방식
- SASL - SASL 프로토콜을 이용한 인증 방식

Note 2. SASL (Simple Authentication and Security Layer) 프레임워크란?

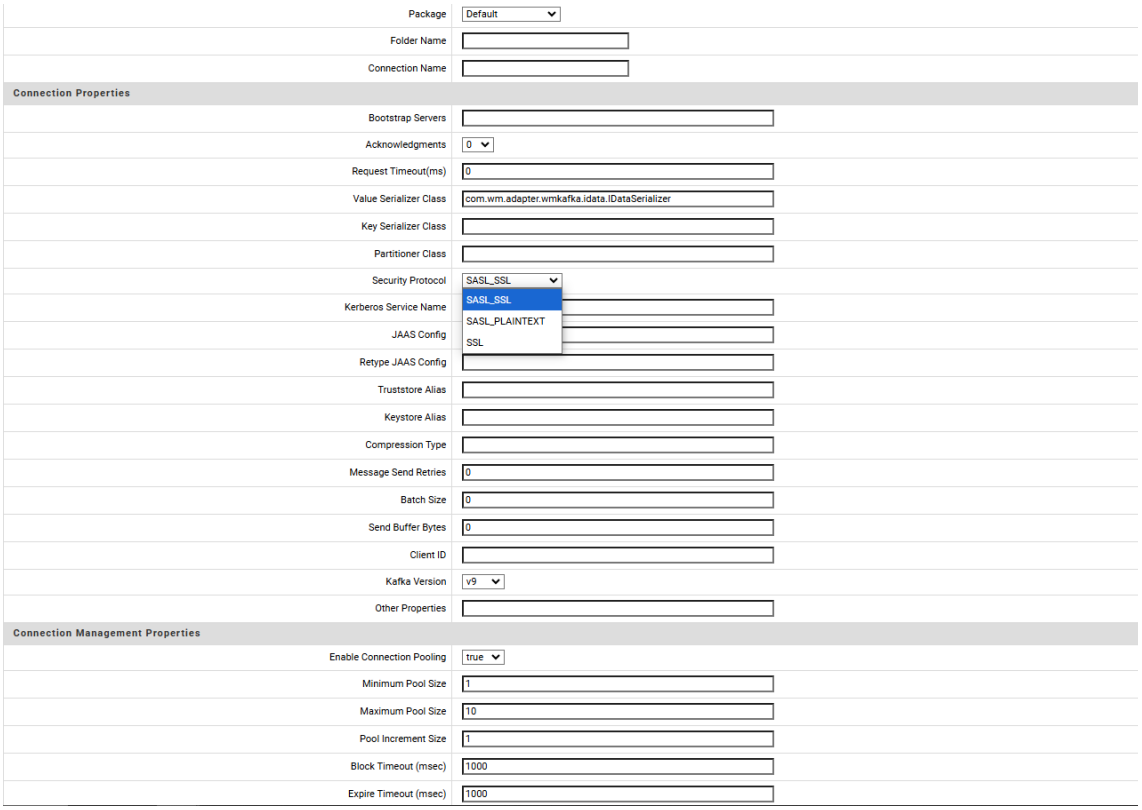


- 연결 지향 프로토콜에서 교체 가능한 메커니즘을 통해 인증, 데이터 보안 서비스를 제공하는 프레임워크
- 인증/인가가 성공 했을 때 후속 데이터 교환이 가능하다.
- 즉, Kafka 가 지원하는 SSL, Kerberos, PLAIN, SCRAM, OAUTH 등 인증 메커니즘을 사용하여 인증/인가 할 수 있도록 함.
  - PLAIN : 문자열 아이디/패스워드를 이용한 인증
  - SCRAM : Salt 등을 이용한 SCRAM 방식을 이용한 인증
  - OAUTHBEARER : OAuth2 방식을 이용한 인증
  - GSSAPI : Kerberos를 이용한 인증

  (webMethods Kafka Adapter 는 SASL_SSL, SASL_PLAINTEXT, SSL 세가지 방식 지원)

※ Apache Kafka SASL Connection

Apache Kafka SASL Producer Connection 설정



Bootstrap Server

webMethods 가 Kafka 내 토픽의 메타데이터를 요청하여 원하는 Broker 를 찾기 위한 서버 주소.

즉 연결 End-point 주소, 일반적으로 IP:Port 로 입력하며, 기본 포트는 9092

Acknowledgements (ALL 설정 권장)

Producer 는 브로커에 데이터를 전송할 때 마다 데이터 쓰기 작업이 완료되었다는 ACK 를 받음.

일반적으로 Cluster 로 구성 된 Kafka 에 다수의 Broker 에 다수의 Partition 으로 토픽이 Replication 되어 있을 경우의 설정함.

0 - 전송과 동시에 성공, Ack 를 받지 않음, 데이터 손실 위험

1 - 전송 후 리더 브로커가 확인 응답을 받을 때, 쓰기가 성공했다고 간주, 데이터 복제 여부 확인 불가능

ALL - 동기화 된 모든 레플리카들이 메시지를 수신하였을 때 메시지 쓰기에 성공했다 간주, Kafka 3.0 이상 기본 값, 리더는 모든 레플리카에서 응답을 받아 Producer 에 전달 **(데이터 보장),** 속도는 가장 느림.

Request Timeout - 요청 타임 아웃

Key Serializer Class & Value Serializer Class

Kafka 로 Produce 할 메시지의 Key, Value 값을 직렬화(Serialize) 하기 위한 클래스 지정

일반적으로 String 값일 때 org.apache.kafka.common.serialization.StringSerializer (참고)

webMethods 의 IData (Document) 형식일 때는 com.wm.adapter.wmkafka.idata.IDataserializer 사용

JAAS Config

JAAS Configuration 파일에서 사용하는 형식으로 SASL 연결을 위한 JAAS 로그인 컨텍스트 매개변수

TruststoreAlias

IS 내 설정 된 Truststore 별칭 (SSL 사용 시)

Keystore Alias

IS 내 설정 된 Keystore 별칭 (SSL 사용 시)

Compression Type

전송 할 Message 에 대한 Compression (압축) 타입 설정으로 Producer 는 Partition 단위로 메시지를 압축

4 가지의 압축 타입을 제공 (gzip, snappy, lz4, zstd) **(참고)**

Type	Compression Ratio	CPU Usage	Compression Speed	Network Bandwidth Usage
gzip	Highest	Highest	Slowest	Lowest
snappy	Medium	Moderate	Moderate	Medium
lz4	Low	Lowest	Fastest	Highest
zstd	Medium	Moderate	Moderate	Medium

Send Buffer Bytes

socket 서버가 사용하는 송수신 버퍼 사이즈, -1 설정 시 OS 기본 설정

Kafka server.properties 내 socket.send.buffer.bytes 로 설정 되어 있음.

Batch Size

메시지를 Compression 할 때 최대 어느 Size 까지 모아서 압축할지 설정, 기본 Byte 단위로 설정

Client ID

Kafka 서버에 전달 할 Client ID 문자열, Kafka 서버에서 요청한 Client 를 구분하기 위해 작성

Kafka Version

Kafka 서버 version 에 따른 설정, 대부분의 서버가 2 버전 이상으로 구성되어 있으므로 v11+ 권장

v9 - 0.9.0.x 버전

v10 - 0.10.0.x 버전

v11+ - 이후 버전

Other Properties

propertyName1=value;propertName2=value 방식으로 커넥션에 사용되는 속성 값들을 입력

Producer Configs 참고

ex) sasl.mechanism=SCRAM-SHA-512

Apache Kafka SASL Consumer Connection 설정

- Return to IBM webMethods Adapter for Apache Kafka Connection Types

Configure Connection Type > IBM webMethods Adapter for Apache Kafka	
Package	Default
Folder Name	
Connection Name	
Connection Properties	
Bootstrap Servers	
groupId	
Client ID	
Key Deserializer Class	
Value Deserializer Class	com.wm.adapter.wmkafka.idata.IDataDeserializer
Deserializer Error Handler	
Enable Auto Commit	false
Auto Commit Interval(ms)	
Security Protocol	SASL_SSL
Kerberos Service Name	
JAAS Config	
Retype JAAS Config	
Truststore Alias	
Keystore Alias	
Session Timeout(ms)	
Auto Offset Reset	earliest
Kafka Version	v11+
Other Properties	
Connection Management Properties	
Enable Connection Pooling	true
Minimum Pool Size	1
Maximum Pool Size	10

Group ID

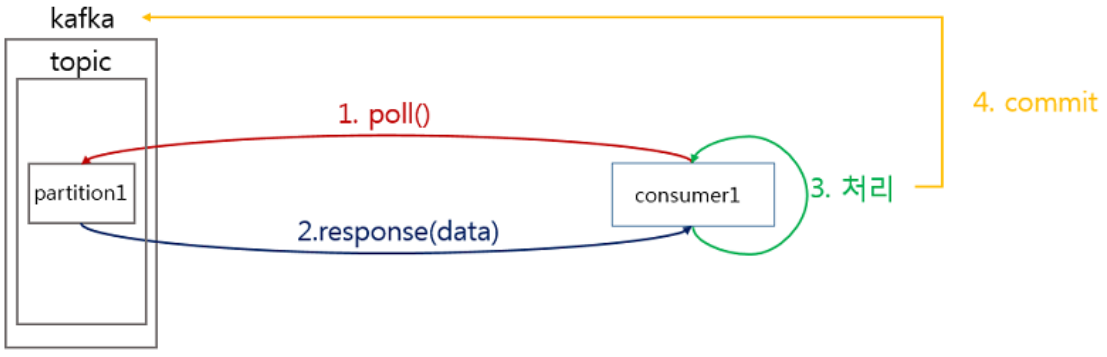
Consumer 는 Group 형태로 동작한다. 같은 컨슈머그룹 안에 속하려면 같은 group.id 값을 갖고 있어야 한다.

Deserializer Error Handler

Kafka Consumer 에서 에러가 발생하였을 때 Error Handling 을 위해 사용하는 Serializer

Enable Auto Commit (**)

Kafka 는 다음과 같은 메시지 Consume 과정을 거침



1. Poll() - Kafka Adapter 의 Consume 요청
2. response(data) - Kafka에 저장 된 메시지를 consumer1 (webMethods) 로 전달
3. Consumer 메시지 처리
4. Offset Commit (Auto Commit 설정에 따라 자동/수동)
    a. 자동 커밋 - Kafka 서버에 설정 된 auto.commit.interval.ms 옵션에 따라 poll() 이후 일정 시간 이후 자동 Offset Commit
    b. 수동 커밋 (commitSync/commitAsync) - Client 에서 Commit 수행, 메시지를 가져오면 서비스 내에서 commitSync(), 또는 commitAsync() 를 통해 Client 가 직접 Offset Commit 을 수행한다.

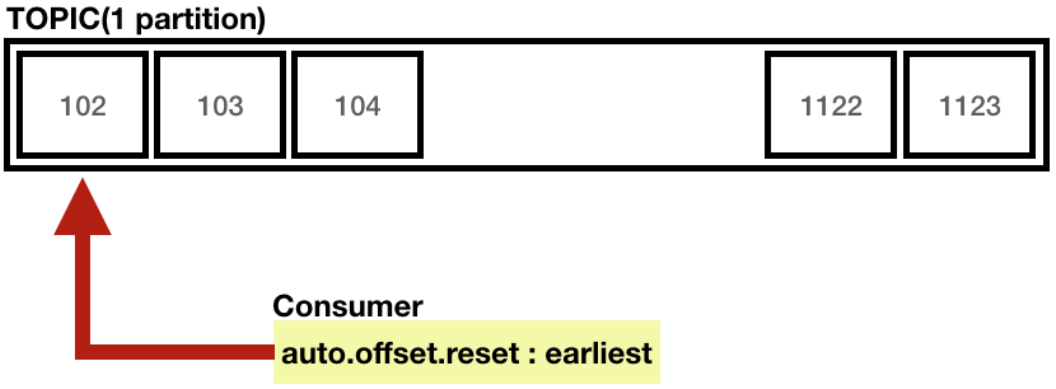
true - Kafka 서버에 설정 된 기본 Auto Commit 설정을 따라 Commit 이 수행된다.

false - Kafka Adapter 수행에 따라 Commit을 수행하며, Adapter Service 수행 시 Consumer Offset 을 Commit 한다.

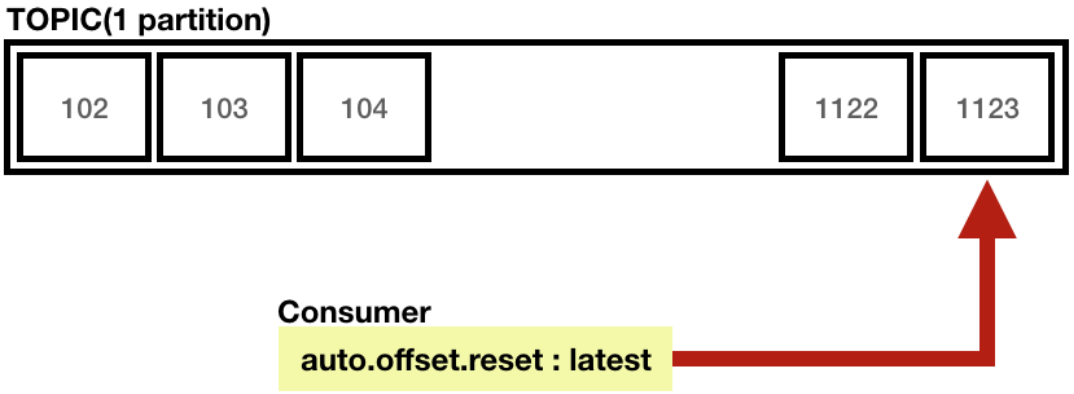
Note. webMethods 의 Kafka Adapter Service 의 수동커밋은 commitSync 를 사용하는 것으로 보임.

Auto Offset Reset(earliest 설정 권장)

Kafka 는 컨슈머 그룹이 읽어간 Offset (Consume 한 메시지까지 위치를 지정하는 일종의 Key 또는 ID 값) 을 __consumer_offset 이라는 토픽에 별도로 저장한다.

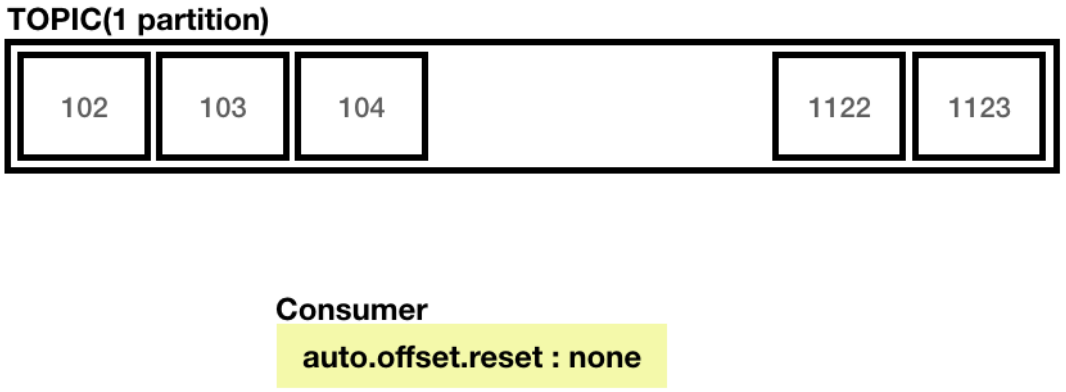


earliest - 마지막 커밋 기록이 없을 경우, 가장 예전(낮은 번호 오프셋) 레코드부터 처리



latest - 마지막 커밋 기록이 없을 경우, 가장 최근(높은 번호 오프셋) 레코드부터 처리 (Default)

→ 리밸런싱 시 데이터의 유실이 발생할 가능성이 있음.



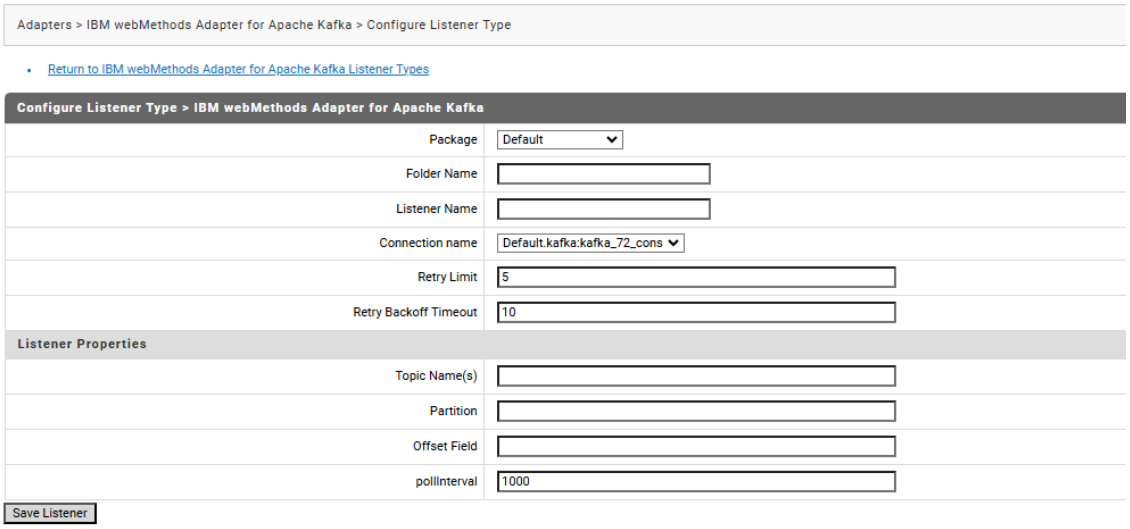
none - 커밋 기록이 없을 경우 throws Exception

Other Properties

propertyName1=value;propertName2=value 방식으로 커넥션에 사용되는 속성 값들을 입력

Consumer Configs 참고

Adapter for Apache Kafka Consumer Listeners 설정



Kafka 로 부터 메시지를 수신하기 위해 사용되는 리스너를 생성.

Consumer Connection 을 매핑하여 생성하며 Kafka에 생성되어 있는 Topic 명을 입력하여 생성

Partition - 토픽 내 파티션을 특정 지을 경우 사용하며, 여러 파티션을 동시에 수신할 경우 Comma(,) 로 분리

Offset Field - 메시지를 수신할 때 Offset 시작점을 설정

Poll Interval - Poll() 을 수행 할 인터벌 설정 (sec)