# Kafka Producer Consumer Developer's Guide

ACI Risk Analytics

Exported on 06/05/2020

# Table of Contents

**Click here to expand TOC**



**Click here to expand Page History**

| Version | Date | Comment |
|---|---|---|
| Current Version[1] **(v. 24)** | **May 04, 2018 08:14** | Aleksey Filiushin[2] |
| v. 23[3] | Apr 13, 2018 12:04 | Diana Kayumova[4] |
| v. 22[5] | Apr 13, 2018 12:03 | Diana Kayumova[6] |
| v. 21[7] | Apr 13, 2018 11:53 | Diana Kayumova[8] |
| v. 20[9] | Apr 13, 2018 11:53 | Diana Kayumova[10] |
| v. 19[11] | Apr 13, 2018 11:49 | Diana Kayumova[12] |
| v. 18[13] | Apr 13, 2018 11:39 | Diana Kayumova[14] |
| v. 17[15] | Apr 13, 2018 11:21 | Diana Kayumova[16] |
| v. 16[17] | Apr 11, 2018 10:45 | Diana Kayumova[18] |

1 https://wiki.aciworldwide.com/display/RMS/viewpage.action?pageId=243781523
2 https://wiki.aciworldwide.com/display/~filiushina
3 https://wiki.aciworldwide.com/display/RMS/viewpage.action?pageId=244668090
4 https://wiki.aciworldwide.com/display/~kayumovad
5 https://wiki.aciworldwide.com/display/RMS/viewpage.action?pageId=243788961
6 https://wiki.aciworldwide.com/display/~kayumovad
7 https://wiki.aciworldwide.com/display/RMS/viewpage.action?pageId=243788957
8 https://wiki.aciworldwide.com/display/~kayumovad
9 https://wiki.aciworldwide.com/display/RMS/viewpage.action?pageId=243788938
10 https://wiki.aciworldwide.com/display/~kayumovad
11 https://wiki.aciworldwide.com/display/RMS/viewpage.action?pageId=243788937
12 https://wiki.aciworldwide.com/display/~kayumovad
13 https://wiki.aciworldwide.com/display/RMS/viewpage.action?pageId=243788931
14 https://wiki.aciworldwide.com/display/~kayumovad
15 https://wiki.aciworldwide.com/display/RMS/viewpage.action?pageId=243788925
16 https://wiki.aciworldwide.com/display/~kayumovad
17 https://wiki.aciworldwide.com/display/RMS/viewpage.action?pageId=243788903
18 https://wiki.aciworldwide.com/display/~kayumovad

| | | |
|---|---|---|
| v. 15[19] | Apr 11, 2018 06:23 | Diana Kayumova[20] |
| v. 14[21] | Apr 10, 2018 10:28 | **user-49ce5** |
| v. 13[22] | Apr 10, 2018 10:22 | Diana Kayumova[23] |
| v. 12[24] | Apr 10, 2018 10:19 | Diana Kayumova[25] |
| v. 11[26] | Apr 10, 2018 10:11 | Diana Kayumova[27] |
| v. 10[28] | Apr 10, 2018 09:15 | Aleksey Filiushin[29] |
| v. 9[30] | Apr 09, 2018 12:09 | **user-49ce5** |
| v. 8[31] | Apr 09, 2018 11:50 | **user-49ce5** |
| v. 7[32] | Apr 09, 2018 01:51 | Aleksey Filiushin[33] |
| v. 6[34] | Apr 06, 2018 12:44 | Diana Kayumova[35] |
| v. 5[36] | Apr 06, 2018 12:41 | Diana Kayumova[37] |
| v. 4[38] | Apr 06, 2018 12:23 | Diana Kayumova[39] |
| v. 3[40] | Apr 06, 2018 12:14 | Diana Kayumova[41] |
| v. 2[42] | Apr 06, 2018 12:08 | Diana Kayumova[43] |

19 https://wiki.aciworldwide.com/display/RMS/viewpage.action?pageId=243786545
20 https://wiki.aciworldwide.com/display/~kayumovad
21 https://wiki.aciworldwide.com/display/RMS/viewpage.action?pageId=243786051
22 https://wiki.aciworldwide.com/display/RMS/viewpage.action?pageId=243785201
23 https://wiki.aciworldwide.com/display/~kayumovad
24 https://wiki.aciworldwide.com/display/RMS/viewpage.action?pageId=243785196
25 https://wiki.aciworldwide.com/display/~kayumovad
26 https://wiki.aciworldwide.com/display/RMS/viewpage.action?pageId=243785194
27 https://wiki.aciworldwide.com/display/~kayumovad
28 https://wiki.aciworldwide.com/display/RMS/viewpage.action?pageId=243785175
29 https://wiki.aciworldwide.com/display/~filiushina
30 https://wiki.aciworldwide.com/display/RMS/viewpage.action?pageId=243785031
31 https://wiki.aciworldwide.com/display/RMS/viewpage.action?pageId=243783820
32 https://wiki.aciworldwide.com/display/RMS/viewpage.action?pageId=243783799
33 https://wiki.aciworldwide.com/display/~filiushina
34 https://wiki.aciworldwide.com/display/RMS/viewpage.action?pageId=243782617
35 https://wiki.aciworldwide.com/display/~kayumovad
36 https://wiki.aciworldwide.com/display/RMS/viewpage.action?pageId=243781577
37 https://wiki.aciworldwide.com/display/~kayumovad
38 https://wiki.aciworldwide.com/display/RMS/viewpage.action?pageId=243781572
39 https://wiki.aciworldwide.com/display/~kayumovad
40 https://wiki.aciworldwide.com/display/RMS/viewpage.action?pageId=243781564
41 https://wiki.aciworldwide.com/display/~kayumovad
42 https://wiki.aciworldwide.com/display/RMS/viewpage.action?pageId=243781551
43 https://wiki.aciworldwide.com/display/~kayumovad

| | | |
|---|---|---|
| v. 1[44] | Apr 06, 2018 11:55 | Diana Kayumova[45] |

44 https://wiki.aciworldwide.com/display/RMS/viewpage.action?pageId=243781538
45 https://wiki.aciworldwide.com/display/~kayumovad

# 1 Introduction

The main goal of this library is to create client side library to simplify interaction with Kafka.

more details at https://kafka.apache.org/documentation/#introduction

more details on implementation see Detail Design document Kafka Producer Consumer Library Design Document[46]

46 https://wiki.aciworldwide.com/display/RMS/Kafka+Producer+Consumer+Library+Design+Document

# 2 Getting started

https://kafka.apache.org/quickstart

Add dependency to your project

```
<dependencies>
    <dependency>
        <groupId>com.aciworldwide.ra</groupId>
        <artifactId>kafka-clients</artifactId>
        <version>1.0.0-SNAPSHOT</version>
    </dependency>
</dependencies>
```

don't forget to add https://nexus02.am.tsacorp.com/content/groups/ra repository into your **settings.xml** file

see usage examples here https://bitbucket02.am.tsacorp.com/projects/TIM/repos/libs/browse/kafka-clients-examples

## 2.1 Producer

1. Create KafkaProducerFactory:

   Kafka requires properties to be set. see at https://kafka.apache.org/documentation/#producerconfigs

2. a. Use property autodetection:
   i. use property file named **`kafka.properties`** `and set producer specific properties there`

   **kafka.properties example**

   ```
   bootstrap.servers=localhost:7090

   #producer specific properties
   key.serializer=org.apache.kafka.common.serialization.StringSerializer
   value.serializer=com.aciworldwide.ra.kafka.serialization.AvroSerializer
   acks=all
   retries=1
   batch.size=16384
   linger.ms=1
   buffer.memory=33554432
   ```

   ii. use system properties with prefix **kafka.** e.g. **-Dkafka.bootstrap.servers=localhost:9092** in both cases use :

```
KafkaProducerFactory factory = KafkaProducerFactory.getInstance();
```

3. Create Producer and pass created Factory as an argument
   there are two type of producers:
   a. MultyTopicProducer
      This producer may be used to send messages to the different topics (as far as they use the same properties). All send() methods of this producer expect topic name as argument.

```
MultiTopicProducer producer = new MultiTopicProducer(factory);
```

   b. FixedTopicProducer
      This producer is intended to send messages to the single/fixed topic. The topic name is set at creation time so client do NOT have to specify it in send() methods.

```
SingleTopicProducer producer = new SingleTopicProducer(factory, TOPIC_NAME);
```

4. Send message using method send
   there are two modes of sending
   a. Synchronous
      Here we wait until message will be delivered to the Kafka broker. If message cannot be delivered send method will return false.
      Be aware: Kafka producer has configurable retry functionality. If we have big number for retry attempts and rather big retry backoff then synchronous call may be blocked for significant amount of time in case of connectivity (or another) issue with the Kafka broker.

```
multiTopicProducer.sendSync(TOPIC_NAME, data);
OR
singleTopicProducer.sendSync(data);
```

   b. Asynchronous
      Here we do not wait any confirmation about delivery of the message to the Kafka broker, so the message could be missed without any notifications.

```
multiTopicProducer.sendAsync(TOPIC_NAME, data);
OR
singleTopicProducer.sendAsync(data);
```

## 2.2 Consumer

1. Create ConsumerBuilder

> Kafka requires properties to be set. see at https://kafka.apache.org/documentation/#consumerconfigs
> There are some extra properties was added:
> **`value.deserializer.output.class`** - the type of the data you are receiving.
> **`consumer.poll.timeout`** - polling timeout. if not set, default value 1000L will be used

2. There are three options to create ConsumerBuilder with properties:
   a. prepare Properties manually and pass it to the builder's **setProperties()** method:

**method chainig ws classical way to set properties**

```
ConsumerBuilder<Data> builder = new ConsumerBuilder<>(myConsumerCallback, TOPIC_NAME)
.setProperties(properties)
.setPollingTimeout(2000L);

or

ConsumerBuilder<Data> builder = new ConsumerBuilder<>(myConsumerCallback, TOPIC_NAME);
builder.setProperties(properties);
builder.setPollingTimeout(2000L);
```

   b. property autodetection:
      i. use property file named **`kafka.properties`** and `set consumer specific properties there`

**kafka.properties example**

```
#consumer specific properties
consumer.poll.timeout=1000
key.deserializer=org.apache.kafka.common.serialization.StringDeserializer
value.deserializer=com.aciworldwide.ra.kafka.serialization.AvroDeserializer
value.deserializer.output.class=test.object.avro.User
auto.offset.reset=earliest
group.id=avro
```

   c. use system properties with prefix **kafka.** e.g. **-Dkafka.bootstrap.servers=localhost:9092**
   in both cases you can omit properties and use "default minimal" configuration in constructor:

```
ConsumerBuilder<Data> builder = new ConsumerBuilder<>(myConsumerCallback, TOPIC_NAME);
Consumer<Data> consumer = builder.build();
```

3. "Default minimal" configuration requires ConsumerCallback to be Implemented. Your business logic goes here.

> Please consider implementing your ConsumerCallback and ConsumerErrorCallback thread safe as it will be called from the thread different than thread where it was created

4. (optional) You can also specify ConsumerErrorCallback for error handling. ConsumerErrorCallback will be called in case of unrecoverable errors.
5. After all configuration set. Create Consumer by calling builder's build method:

6. 
```
Consumer<Data> consumer = builder.build();
```

7. to start receiving a messages call listen method

```
consumer.listen();
```

8. There is also method to stop listening and shutdown consumer. You cannot re-enable the Consumer after calling this method. you need to create a new one.

```
consumer.shutdown()
```

9. To check if consumer is Alive or not.

> It returns
> **`true`** - when consumer is working,
> **`false`** - in all other cases including consumer is not started or closed

```
consumer.isAlive()
```

## 2.3 Avro serialization

The library also provides Avro serializer and deserializer for Kafka:

- `com.aciworldwide.ra.kafka.serialization.AvroSerializer`
- `com.aciworldwide.ra.kafka.serialization.AvroDeserializer`
  The Deserializer should be aware about the target class which should be used for deserialization.
  This why it requires additional property to be set: "value.deserializer.output.class".
  The property accept both fully classified class name of class (as string) and Class<?> object.

You can use them by specifying corresponding Kafka properties:

- "value.serializer" for producer
- "value.deserializer" and for consumer

Avro schema example

**user.avsc**

```
{"namespace": "test.object.avro",
  "type": "record",
  "name": "User",
  "fields": [
    {"name": "name", "type": "string"},
    {"name": "favorite_number",  "type": ["int", "null"]},
    {"name": "favorite_color", "type": ["string", "null"]}
  ]
}
```

you can generate sources from schema using avro-maven-plugin. e.g.:

**pom.xml**

```
<build>
    <plugins>
        <!-- avro-maven-plugin -->
        <plugin>
            <groupId>org.apache.avro</groupId>
            <artifactId>avro-maven-plugin</artifactId>
            <version>1.8.2</version>
            <executions>
                <execution>
                    <phase>generate-sources</phase>
                    <goals>
                        <goal>schema</goal>
                    </goals>
                </execution>
            </executions>
        </plugin>
    </plugins>
</build>
```

# 3 Security

Our current expectation is that Kafka will be used to transfer some kind of sensitive information.
To be compliant with security standards we should protect such data at Rest(DaRE) and in Transite (DiTE). The DaRE part should be covered by broker configuration and this why outside of the scope of this library.
With the library you can use all means provided by Kafka to secure communication, including Data encryption (TLS) and authentication (to limit access to the Kafka).
To turn on all of this means just pass corresponding properties to the Producer/Consumer.
For property details see: https://kafka.apache.org/documentation/#security_overview