AWS-DDBSTREAM

DynamoDB Stream Component

Available as of Camel 2.7

The DynamoDB Stream component supports receiving messages from Amazon DynamoDB Stream service.

> ⓘ **Prerequisites**
>
> You must have a valid Amazon Web Services developer account, and be signed up to use Amazon DynamoDB Streams. More information are available at AWS DynamoDB

URI Format

```
aws-ddbstream://table-name[?options]
```

The stream needs to be created prior to it being used.
You can append query options to the URI in the following format, ?options=value&option2=value&...

URI Options

Name	Default Value	Context	Description
amazonDynamoDbStreamsClient	`null`	Consumer	Reference to a `com.amazonaws.services.kinesis.AmazonDynamoDBStreams` in the Registry.
maxMessagesPerPoll	`100`	Consumer	Maximum results that will be returned in each poll to the AWS API, Given that the shard iterator is unique to the consumer, changing it shouldn't effect other consumers.
iteratorType	`LATEST`	Consumer	One of `trim_horizon`, `latest`, `after_sequence_number`, or `at_sequence_number`. See http://docs.aws.amazon.com/dynamodbstreams/latest/APIReference/API_GetShardIterator.html for descriptions of these two iterator types.
sequenceNumberProvider	null	Consumer	Either a bean reference to an implementation of `org.apache.camel.component.aws.ddbstream.SequenceNumberProvider` or a literal string representing a sequence number. The role of this is to determine where in the stream to start when using one of the `after_sequence_number`, or `at_sequence_number` iterator types.

> ⓘ **Required DynampDBStream component options**
>
> You have to provide the amazonDynamoDbStreamsClient in the Registry with proxies and relevant credentials configured.

Sequence Numbers

You can provide a literal string as the sequence number or provide a bean in the registry. An example of using the bean would be to save your current position in the change feed and restore it on Camel startup.

It is an error to provide a sequence number that is greater than the largest sequence number in the describe-streams result, as this will lead to the AWS call returning an HTTP 400.

Batch Consumer

This component implements the Batch Consumer.

This allows you for instance to know how many messages exists in this batch and for instance let the Aggregator aggregate this number of messages.

Usage

AmazonDynamoDBStreamsClient configuration

You will need to create an instance of AmazonDynamoDBStreamsClient and bind it to the registry

```
ClientConfiguration clientConfiguration = new ClientConfiguration();
clientConfiguration.setProxyHost("http://myProxyHost");
clientConfiguration.setProxyPort(8080);

Region region = Region.getRegion(Regions.fromName(region));
region.createClient(AmazonDynamoDBStreamsClient.class, null, clientConfiguration);
// the 'null' here is the AWSCredentialsProvider which defaults to an instance of
DefaultAWSCredentialsProviderChain

registry.bind("kinesisClient", client);
```

Providing AWS Credentials

It is recommended that the credentials are obtained by using the DefaultAWSCredentialsProviderChain that is the default when creating a new ClientConfiguration instance, however, a different AWSCredentialsProvider can be specified when calling createClient(...).

Coping with Downtime

AWS DynamoDB Streams outage of less than 24 hours

The consumer will resume from the last seen sequence number (as implemented for CAMEL-9515), so you should receive a flood of events in quick succession, as long as the outage did not also include DynamoDB itself.

AWS DynamoDB Streams outage of more than 24 hours

Given that AWS only retain 24 hours worth of changes, you will have missed change events no matter what mitigations are in place.

Dependencies

Maven users will need to add the following dependency to their pom.xml.

pom.xml

```
<dependency>
    <groupId>org.apache.camel</groupId>
    <artifactId>camel-aws</artifactId>
    <version>${camel-version}</version>
</dependency>
```

where ${camel-version} must be replaced by the actual version of Camel (2.7 or higher).

See Also

- Configuring Camel
- Component
- Endpoint
- Getting Started

- AWS Component