AWS-SQS

SQS Component

Available as of Camel 2.6

The sqs component supports sending and receiving messages to Amazon's SQS service.

Prerequisites

You must have a valid Amazon Web Services developer account, and be signed up to use Amazon SQS. More information are available at Amazon SQS.

URI Format

```
aws-sqs://queueName[?options]
aws-sqs://queueNameOrArn[?options] (from Camel 2.18)
```

The queue will be created if they don't already exists. You can append query options to the URI in the following format: **?options=value&option2=value&...**

URI Options

Name	Default Value	Context	Description
accessKey	null	Shared	Amazon AWS Access Key.
amazonSQSClient	null	Shared	Reference to a **com.amazonaws.services.sqs.AmazonSQS** in the Registry.
amazonSQSEndpoint	null	Shared	The region with which the **aws-sqs** client wants to work with. Only works if Camel creates the **aws-sqs** client, i.e., if you explicitly set **amazonSQSClient**, then this setting will have no effect. You would have to set it on the client you create directly
attributeNames	null	Consumer	A list of attribute names to receive when consuming. **Camel 2.17:** Multiple names can be separated by comma. **Camel 2.16** or older: The type is a Collection so its much harder to configure and use.
concurrentConsumers	1	Consumer	(**as of 2.15.0**) Allows you to use multiple threads to poll the SQS queue to increase throughput. You must also set the **maxMessagesPerPoll** option for this to work properly.
defaultVisibilityTimeout	null	Shared	The visibility timeout (in seconds) to set in the **com.amazonaws.services.sqs.model.CreateQueueRequest**.
delaySeconds	null	Producer	**Camel 2.9.3:** Delay sending messages for a number of seconds.
deleteAfterRead	true	Consumer	Delete message from SQS after it has been read (and processed by the route). If this option is false, then the same objects will be retrieve over and over again on the polls. Therefore you need to use the Idempotent Consumer EIP in the route to filter out duplicates. You can filter using the **S3Constants#BUCKET_NAME** and **S3Constants#KEY** headers, or only the **S3Constants#KEY** header.
deleteIfFiltered	true	Consumer	**Camel 2.12.2, 2.13.0**: Whether or not to send the **DeleteMessage** to the SQS queue if an exchange fails to get through a filter. If **false** and exchange does not make it through a Camel filter upstream in the route, then don't send **DeleteMessage**.
extendMessageVisibility	false	Consumer	**Camel 2.10:** If enabled a scheduled background task will keep extending the message visibility on SQS. This is needed if it takes a long time to process the message. If set to true **visibilityTimeout** must be set. See details at Amazon docs.
maximumMessageSize	null	Shared	**Camel 2.8:** The **maximumMessageSize** (in bytes) an SQS message can contain for this queue, to set in the **com.amazonaws.services.sqs.model.SetQueueAttributesRequest**.
maxMessagesPerPoll	null	Consumer	The maximum number of messages which can be received in one poll to set in the **com.amazonaws.services.sqs.model.ReceiveMessageRequest**.
messageAttributeNames	null	Consumer	A list of message attribute names to receive when consuming. **Camel 2.17:** Multiple names can be separated by comma. **Camel 2.16** or older: The type is a Collection so its much harder to configure and use.

messageRetentionPeriod	null	Shared	**Camel 2.8:** The `messageRetentionPeriod` (in seconds) a message will be retained by SQS for this queue, to set in the `com.amazonaws.services.sqs.model.SetQueueAttributesRequest`.
proxyHost	null	Shared	**Camel 2.16**: Specify a proxy host to be used inside the client definition.
proxyPort	null	Shared	**Camel 2.16**: Specify a proxy port to be used inside the client definition.
queueOwnerAWSAccountId	null	Shared	**Camel 2.12:** Specify the queue owner aws account id when you need to connect the queue with different account owner.
policy	null	Shared	**Camel 2.8:** The policy for this queue to set in the `com.amazonaws.services.sqs.model.SetQueueAttributesRequest`.
receiveMessageWaitTimeSeconds	0	Shared	**Camel 2.11:** If you do not specify `WaitTimeSeconds` in the request, the queue attribute `ReceiveMessageWaitTimeSeconds` is used to determine how long to wait.
redrivePolicy	null	Shared	**Camel 2.15**: Specify the policy that send message to `DeadLetter` queue. See detail at Amazon docs.
region	null	Shared	**Camel 2.12.3:** Specify the queue region which could be used with `queueOwnerAWSAccountId` to build the service URL. **Note**: Region will still default to **us-east-1** if `queueOwnerAWSAccountId` is not specified
secretKey	null	Shared	Amazon AWS Secret Key.
waitTimeSeconds	0	Producer	**Camel 2.11:** Duration in seconds (0 to 20) that the `ReceiveMessage` action call will wait until a message is in the queue to include in the response.
visibilityTimeout	null	Shared	The duration (in seconds) that the received messages are hidden from subsequent retrieve requests after being retrieved by a `ReceiveMessage` request. This only make sense if its different from `defaultVisibilityTimeout`.

Required SQS component options

You have to provide the `amazonSQSClient` in the Registry or your `accessKey` and `secretKey` to access the Amazon's SQS.

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

Message headers set by the SQS producer

Header	Type	Description
`CamelAwsSqsMD5OfBody`	String	The MD5 checksum of the Amazon SQS message.
`CamelAwsSqsMessageId`	String	The Amazon SQS message ID.
`CamelAwsSqsDelaySeconds`	Integer	Since **Camel 2.11**, the delay seconds that the Amazon SQS message can be see by others.

Message headers set by the SQS consumer

Header	Type	Description
`CamelAwsSqsMD5OfBody`	String	The MD5 checksum of the Amazon SQS message.
`CamelAwsSqsMessageId`	String	The Amazon SQS message ID.
`CamelAwsSqsReceiptHandle`	String	The Amazon SQS message receipt handle.
`CamelAwsSqsAttributes`	Map<String, String>	The Amazon SQS message attributes.

Advanced AmazonSQS configuration

If your Camel Application is running behind a firewall or if you need to have more control over the AmazonSQS instance configuration, you can create your own instance:

```
AWSCredentials awsCredentials = new BasicAWSCredentials("myAccessKey", "mySecretKey");

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

AmazonSQS client = new AmazonSQSClient(awsCredentials, clientConfiguration);

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

and refer to it in your Camel aws-sqs component configuration:

```
from("aws-sqs://MyQueue?amazonSQSClient=#client&delay=5000&maxMessagesPerPoll=5")
   .to("mock:result");
```

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 (2.6 or higher).

JMS-style Selectors

SQS does not allow selectors, but you can effectively achieve this by using the Camel Filter EIP and setting an appropriate **visibilityTimeout**. When SQS dispatches a message, it will wait up to the visibility timeout before it will try to dispatch the message to a different consumer unless a **DeleteMessage** is received. By default, Camel will always send the **DeleteMessage** at the end of the route, unless the route ended in failure. To achieve appropriate filtering and not send the **DeleteMessage** even on successful completion of the route, use a Filter:

```
from("aws-sqs://MyQueue?amazonSQSClient=#client&defaultVisibilityTimeout=5000&deleteIfFiltered=false")
   .filter("${header.login} == true")
   .to("mock:result");
```

In the above code, if an exchange doesn't have an appropriate header, it will not make it through the filter AND also not be deleted from the SQS queue. After 5000 miliseconds, the message will become visible to other consumers.

See Also

- Configuring Camel
- Component
- Endpoint
- Getting Started

- AWS Component