

**Qualcomm Car-to-Cloud Platform**

**QUEUE IMPLEMENTATION DOCUMENT**

**Version No.2.0**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Prepared By / Last Updated By** | **Reviewed By** | **Approved By** |
| **Name** | Soumya | Sudheer |  |
| **Role** | PAT | Sr Manager |  |
| **Signature** |  |  |  |
| **Date** | May 18,2021 | May 20,2021 |  |

May 18, 2021

Cognizant

Table of Contents

[1. MESSAGE QUEUE IMPLEMENTATION 2](#_Toc72415985)

[1.1. The Dependencies 2](#_Toc72415986)

[1.2. The IC2CServiceQueueClient Bean Creation 2](#_Toc72415987)

[1.3. The Publish Method 3](#_Toc72415988)

[1.4. The Receive Method 3](#_Toc72415989)

[1.5. The Listen Method 4](#_Toc72415990)

[1.6. The Delete Message Method 4](#_Toc72415991)

[1.7. The Enable Listener Method 4](#_Toc72415992)

[1.8. The Disable Listener Method 5](#_Toc72415993)

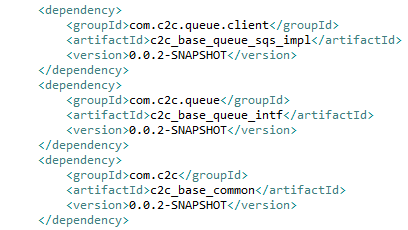
[1.9. The Reconnect Method 5](#_Toc72415994)

# MESSAGE QUEUE IMPLEMENTATION

To use the AWS SQS implementation, find the below steps to implement AWS SQS functions.

## The Dependencies

Create the jars of both the interface and the implementation project. Add these jars to the pom.xml file of the caller project. The required dependencies that need to be added are:

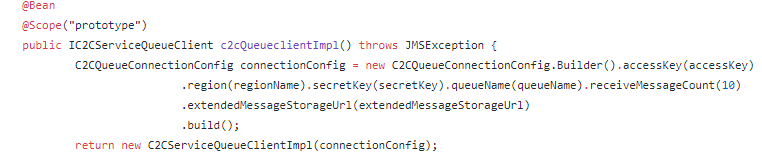


## The IC2CServiceQueueClient Bean Creation

Create an object of **C2CQueueConnectionConfig** with the required parameters queue name, access key, secret key and region name.

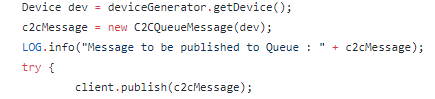
Create a bean of the **IC2CServiceQueueClient** with the bean scope as **prototype** where the C2CQueueConnectionConfig object is injected as a dependency to it**.**

The queue name, access key, secret key and region name will be taken from external properties file.



## The Publish Method

Using the **IC2CServiceQueueClient** object, call the publish method to send messages to the queue. We pass an object of **C2CQueueMessage** class as a parameter for the publish method.



***Exceptions:***

**QueueValidationException** is thrown when the message size is greater than 1 MB.

**QueueValidationException** is also thrown when the **C2CQueueMessage** is not serialized and is returning a null value.

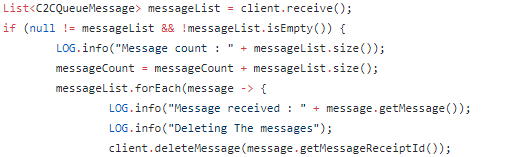
**InvalidQueueException** is thrown when an AmazonSQS Exception or a BaseCommonApplicationException occurs.

## The Receive Method

Using the **IC2CServiceQueueClient** object, call the receive method which returns a list of **C2CQueueMessage.**

Once the message is received, the caller application should explicitly delete the message using the **deleteMessage** exposed in the interface.

Pass the messageReceiptId as a parameter to the delete method.



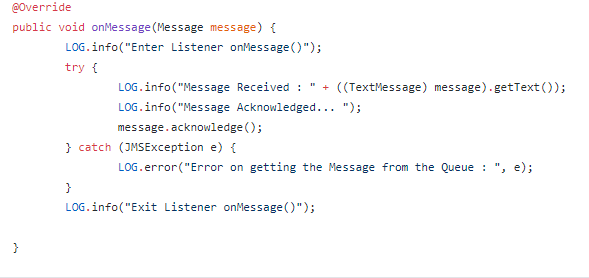
***Exceptions:***

**InvalidQueueException** is thrown when an AmazonSQS Exception occurs.

## The Listen Method

Implement the JMS listener interface. The onMessage method of the MessageListener interface is called when we receive a message.

Using the **IC2CServiceQueueClient** object, call the listen method of the **C2C\_Base\_Message\_Broker\_SQS\_Impl** and pass the instance of the listener implementation as parameter. Explicitly acknowledge the messages after they are received.



## The Delete Message Method

Using the **IC2CServiceQueueClient** object, call the delete method to delete messages from the queue. We pass the messageReceiptId as a parameter to the delete method.

***Exceptions:***

**InvalidQueueException** is thrown when an AmazonSQS Exception occurs.

## The Enable Listener Method

Using the **IC2CServiceQueueClient** object, call the enableListener method to enable the listener to consume messages from the queue. This method will start listening to a specific Queue and the Consumer will start receiving the messages.



## The Disable Listener Method

Using the **IC2CServiceQueueClient** object, call the disableListener method to disable the listener to consume messages from the queue. This method will stop listening to a specific Queue and the Consumer will no longer receive any message.



## The Reconnect Method

Using the **IC2CServiceQueueClient** object bean, call the reconnect method to reconnect the client for connection establishment.

The caller application should invoke the publish method after reconnecting and listener method to listen to the queue.

