aws

API Reference

Amazon Simple Queue Service



API Version 2012-11-05

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon Simple Queue Service: API Reference

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

Welcome **1**
Actions **2**
AddPermission 3
Request Syntax 3
Request Parameters 4
Response Elements 5
Errors 5
Examples 6
See Also 7
CancelMessageMoveTask 9
Request Syntax 9
Request Parameters 9
Response Syntax 9
Response Elements 10
Errors 10
Examples 11
See Also 12
ChangeMessageVisibility 13
Request Syntax 14
Request Parameters 14
Response Elements 15
Errors 15
Examples 16
See Also 18
ChangeMessageVisibilityBatch 19
Request Syntax 19
Request Parameters 19
Response Syntax 20
Response Elements 20
Errors 21
Examples 22
See Also 25
CreateQueue 26
Request Syntax 27

Request Parameters ... 27
Response Syntax ... 32
Response Elements ... 32
Errors ... 32
Examples ... 34
See Also ... 36
DeleteMessage ... 37
Request Syntax ... 37
Request Parameters ... 37
Response Elements ... 38
Errors ... 38
Examples ... 39
See Also ... 41
DeleteMessageBatch ... 42
Request Syntax ... 42
Request Parameters ... 42
Response Syntax ... 43
Response Elements ... 43
Errors ... 44
Examples ... 45
See Also ... 47
DeleteQueue ... 49
Request Syntax ... 49
Request Parameters ... 49
Response Elements ... 50
Errors ... 50
Examples ... 51
See Also ... 52
GetQueueAttributes ... 53
Request Syntax ... 53
Request Parameters ... 53
Response Syntax ... 57
Response Elements ... 58
Errors ... 58
Examples ... 59
See Also ... 64

GetQueueUrl ... 65
Request Syntax ... 65
Request Parameters ... 65
Response Syntax ... 66
Response Elements ... 66
Errors ... 66
Examples ... 67
See Also ... 68
ListDeadLetterSourceQueues ... 70
Request Syntax ... 70
Request Parameters ... 70
Response Syntax ... 71
Response Elements ... 71
Errors ... 72
Examples ... 72
See Also ... 74
ListMessageMoveTasks ... 75
Request Syntax ... 75
Request Parameters ... 75
Response Syntax ... 76
Response Elements ... 76
Errors ... 76
Examples ... 77
See Also ... 79
ListQueues ... 80
Request Syntax ... 80
Request Parameters ... 80
Response Syntax ... 81
Response Elements ... 81
Errors ... 82
Examples ... 82
See Also ... 84
ListQueueTags ... 86
Request Syntax ... 86
Request Parameters ... 86
Response Syntax ... 86

Response Elements ... 87
Errors ... 87
Examples ... 88
See Also ... 89
PurgeQueue ... 91
Request Syntax ... 91
Request Parameters ... 91
Response Elements ... 92
Errors ... 92
Examples ... 93
See Also ... 94
ReceiveMessage ... 95
Request Syntax ... 96
Request Parameters ... 96
Response Syntax ... 102
Response Elements ... 102
Errors ... 102
Examples ... 104
See Also ... 109
RemovePermission ... 110
Request Syntax ... 110
Request Parameters ... 110
Response Elements ... 111
Errors ... 111
Examples ... 112
See Also ... 113
SendMessage ... 114
Request Syntax ... 114
Request Parameters ... 115
Response Syntax ... 119
Response Elements ... 119
Errors ... 120
Examples ... 122
See Also ... 125
SendMessageBatch ... 127
Request Syntax ... 127

Request Parameters ... 128
Response Syntax ... 129
Response Elements ... 129
Errors ... 130
Examples ... 132
See Also ... 136
SetQueueAttributes ... 138
Request Syntax ... 138
Request Parameters ... 138
Response Elements ... 142
Errors ... 142
Examples ... 144
See Also ... 152
StartMessageMoveTask ... 153
Request Syntax ... 153
Request Parameters ... 153
Response Syntax ... 154
Response Elements ... 154
Errors ... 155
Examples ... 155
See Also ... 156
TagQueue ... 158
Request Syntax ... 158
Request Parameters ... 158
Response Elements ... 159
Errors ... 159
Examples ... 160
See Also ... 161
UntagQueue ... 163
Request Syntax ... 163
Request Parameters ... 163
Response Elements ... 164
Errors ... 164
Examples ... 164
See Also ... 166
Data Types ... 167

BatchResultErrorEntry ... 168
Contents ... 168
See Also ... 168
ChangeMessageVisibilityBatchRequestEntry ... 170
Contents ... 170
See Also ... 170
ChangeMessageVisibilityBatchResultEntry ... 172
Contents ... 172
See Also ... 172
DeleteMessageBatchRequestEntry ... 173
Contents ... 173
See Also ... 173
DeleteMessageBatchResultEntry ... 174
Contents ... 174
See Also ... 174
ListMessageMoveTasksResultEntry ... 175
Contents ... 175
See Also ... 177
Message ... 178
Contents ... 178
See Also ... 180
MessageAttributeValue ... 181
Contents ... 181
See Also ... 182
MessageSystemAttributeValue ... 183
Contents ... 183
See Also ... 184
SendMessageBatchRequestEntry ... 185
Contents ... 185
See Also ... 189
SendMessageBatchResultEntry ... 190
Contents ... 190
See Also ... 191
Common Parameters ... 192
Common Errors ... 195

Welcome

Welcome to the *Amazon SQS API Reference*.

Amazon SQS is a reliable, highly-scalable hosted queue for storing messages as they travel between applications or microservices. Amazon SQS moves data between distributed application components and helps you decouple these components.

For information on the permissions you need to use this API, see Identity and access management in the *Amazon SQS Developer Guide.*

You can use AWS SDKs to access Amazon SQS using your favorite programming language. The SDKs perform tasks such as the following automatically:

- Cryptographically sign your service requests
- Retry requests
- Handle error responses

Additional information

- Amazon SQS Product Page
- *Amazon SQS Developer Guide*
  - Making API Requests
  - Amazon SQS Message Attributes
  - Amazon SQS Dead-Letter Queues
- Amazon SQS in the AWS Command Line Interface
- *Amazon Web Services General Reference*
  - Regions and Endpoints

This document was last published on November 14, 2026.

Actions

The following actions are supported:

- AddPermission
- CancelMessageMoveTask
- ChangeMessageVisibility
- ChangeMessageVisibilityBatch
- CreateQueue
- DeleteMessage
- DeleteMessageBatch
- DeleteQueue
- GetQueueAttributes
- GetQueueUrl
- ListDeadLetterSourceQueues
- ListMessageMoveTasks
- ListQueues
- ListQueueTags
- PurgeQueue
- ReceiveMessage
- RemovePermission
- SendMessage
- SendMessageBatch
- SetQueueAttributes
- StartMessageMoveTask
- TagQueue
- UntagQueue

AddPermission

Adds a permission to a queue for a specific principal. This allows sharing access to the queue.

When you create a queue, you have full control access rights for the queue. Only you, the owner of the queue, can grant or deny permissions to the queue. For more information about these permissions, see Allow Developers to Write Messages to a Shared Queue in the *Amazon SQS Developer Guide*.

Note

- `AddPermission` generates a policy for you. You can use `SetQueueAttributes` to upload your policy. For more information, see Using Custom Policies with the Amazon SQS Access Policy Language in the *Amazon SQS Developer Guide*.
- An Amazon SQS policy can have a maximum of seven actions per statement.
- To remove the ability to change queue permissions, you must deny permission to the `AddPermission`, `RemovePermission`, and `SetQueueAttributes` actions in your IAM policy.
- Amazon SQS `AddPermission` does not support adding a non-account principal.

Note

Cross-account permissions don't apply to this action. For more information, see Grant cross-account permissions to a role and a username in the *Amazon SQS Developer Guide*.

Request Syntax

```
{
   "Actions": [ "string" ],
   "AWSAccountIds": [ "string" ],
   "Label": "string",
   "QueueUrl": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

Actions

The action the client wants to allow for the specified principal. Valid values: the name of any action or `*`.

For more information about these actions, see Overview of Managing Access Permissions to Your Amazon Simple Queue Service Resource in the *Amazon SQS Developer Guide*.

Specifying `SendMessage`, `DeleteMessage`, or `ChangeMessageVisibility` for `ActionName.n` also grants permissions for the corresponding batch versions of those actions: `SendMessageBatch`, `DeleteMessageBatch`, and `ChangeMessageVisibilityBatch`.

Type: Array of strings

Required: Yes

AWSAccountIds

The AWS account numbers of the principals who are to receive permission. For information about locating the AWS account identification, see Your AWS Identifiers in the *Amazon SQS Developer Guide*.

Type: Array of strings

Required: Yes

Label

The unique identification of the permission you're setting (for example, `AliceSendMessage`). Maximum 80 characters. Allowed characters include alphanumeric characters, hyphens (`-`), and underscores (`_`).

Type: String

Required: Yes

QueueUrl

The URL of the Amazon SQS queue to which permissions are added.

Queue URLs and names are case-sensitive.

Type: String

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors.

InvalidAddress

The specified ID is invalid.

HTTP Status Code: 400

InvalidSecurity

The request was not made over HTTPS or did not use SigV4 for signing.

HTTP Status Code: 400

OverLimit

The specified action violates a limit. For example, `ReceiveMessage` returns this error if the maximum number of in flight messages is reached and `AddPermission` returns this error if the maximum number of permissions for the queue is reached.

HTTP Status Code: 400

QueueDoesNotExist

Ensure that the `QueueUrl` is correct and that the queue has not been deleted.

HTTP Status Code: 400

RequestThrottled

The request was denied due to request throttling.

- Exceeds the permitted request rate for the queue or for the recipient of the request.
- Ensure that the request rate is within the Amazon SQS limits for sending messages. For more information, see Amazon SQS quotas in the *Amazon SQS Developer Guide*.

HTTP Status Code: 400

UnsupportedOperation

Error code 400. Unsupported operation.

HTTP Status Code: 400

Examples

The following example query requests grant a `SendMessage` permission to the principal whose AWS account number is 177715257436 and a `ReceiveMessage` permission to the principal whose AWS account number is 111111111111. The structure of AUTHPARAMS depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the *AWS General Reference*.

Example

Using AWS JSON protocol (Default)

Sample Request

```
POST / HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Target: AmazonSQS.AddPermission
X-Amz-Date: <Date>
Content-Type: application/x-amz-json-1.0
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
{
    "QueueUrl": "https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue/",
    "Label": "MyLabel",
    "Actions": ["SendMessage", "ReceiveMessage"],
    "AWSAccountIds": ["177715257436", "111111111111"]
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <requestId>
Content-Length: 0
Date: <Date>
Content-Type: application/x-amz-json-1.0
```

Example

Using AWS query protocol

Sample Request

```
POST /177715257436/MyQueue HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Date: <Date>
Content-Type: application/x-www-form-urlencoded
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
Action=AddPermission
&Label=MyLabel
&AWSAccountId.1=177715257436
&ActionName.1=SendMessage
&AWSAccountId.2=111111111111
&ActionName.2=ReceiveMessage
```

Sample Response

```
HTTP/1.1 200 OK
<?xml version="1.0"?>
<AddPermissionResponse xmlns="http://queue.amazonaws.com/doc/2012-11-05/">
    <ResponseMetadata>
        <RequestId>b05e862e-c961-5213-bd8d-da5e6667b7d7</RequestId>
    </ResponseMetadata>
</AddPermissionResponse>
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python (Boto3)
- AWS SDK for Ruby V3

CancelMessageMoveTask

Cancels a specified message movement task. A message movement can only be cancelled when the current status is RUNNING. Cancelling a message movement task does not revert the messages that have already been moved. It can only stop the messages that have not been moved yet.

Note

- This action is currently limited to supporting message redrive from dead-letter queues (DLQs) only. In this context, the source queue is the dead-letter queue (DLQ), while the destination queue can be the original source queue (from which the messages were driven to the dead-letter-queue), or a custom destination queue.
- Only one active message movement task is supported per queue at any given time.

Request Syntax

```
{
   "TaskHandle": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

TaskHandle

An identifier associated with a message movement task.

Type: String

Required: Yes

Response Syntax

```
{
```

```
    "ApproximateNumberOfMessagesMoved": number
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ApproximateNumberOfMessagesMoved

The approximate number of messages already moved to the destination queue.

Type: Long

Errors

For information about the errors that are common to all actions, see Common Errors.

InvalidAddress

The specified ID is invalid.

HTTP Status Code: 400

InvalidSecurity

The request was not made over HTTPS or did not use SigV4 for signing.

HTTP Status Code: 400

RequestThrottled

The request was denied due to request throttling.

- Exceeds the permitted request rate for the queue or for the recipient of the request.
- Ensure that the request rate is within the Amazon SQS limits for sending messages. For more information, see Amazon SQS quotas in the *Amazon SQS Developer Guide*.

HTTP Status Code: 400

ResourceNotFoundException

One or more specified resources don't exist.

HTTP Status Code: 400

UnsupportedOperation

Error code 400. Unsupported operation.

HTTP Status Code: 400

Examples

Example

Using AWS query protocol

The following example query cancels an existing running message move task. The structure of AUTHPARAMS depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the *AWS General Reference*.

Sample Request

```
POST / HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Date: <Date>
Content-Type: application/x-www-form-urlencoded
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
Action=CancelMessageMoveTask
&TaskHandle=eyJ0YXNrSWQiOiJkYzE2OWUwNC0wZTU1LTQ0ZDItYWE5MC1jMDgwY2ExZjM2ZjciLCJzb3VyY2VBcm4iOiJ
```

Sample Response

```
HTTP/1.1 200 OK
<?xml version="1.0"?>
<CancelMessageMoveTaskResponse xmlns="http://queue.amazonaws.com/doc/2012-11-05/">
    <CancelMessageMoveTaskResult>
        <ApproximateNumberOfMessagesMoved>300</ApproximateNumberOfMessagesMoved>
    </CancelMessageMoveTaskResult>
    <ResponseMetadata>
        <RequestId>9b20926c-8b35-5d8e-9559-ce1c22e754dc</RequestId>
    </ResponseMetadata>
</CancelMessageMoveTaskResponse>
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python (Boto3)
- AWS SDK for Ruby V3

ChangeMessageVisibility

Changes the visibility timeout of a specified message in a queue to a new value. The default visibility timeout for a message is 30 seconds. The minimum is 0 seconds. The maximum is 12 hours. For more information, see Visibility Timeout in the *Amazon SQS Developer Guide*.

For example, if the default timeout for a queue is 60 seconds, 15 seconds have elapsed since you received the message, and you send a ChangeMessageVisibility call with `VisibilityTimeout` set to 10 seconds, the 10 seconds begin to count from the time that you make the `ChangeMessageVisibility` call. Thus, any attempt to change the visibility timeout or to delete that message 10 seconds after you initially change the visibility timeout (a total of 25 seconds) might result in an error.

An Amazon SQS message has three basic states:

1. Sent to a queue by a producer.
2. Received from the queue by a consumer.
3. Deleted from the queue.

A message is considered to be *stored* after it is sent to a queue by a producer, but not yet received from the queue by a consumer (that is, between states 1 and 2). There is no limit to the number of stored messages. A message is considered to be *in flight* after it is received from a queue by a consumer, but not yet deleted from the queue (that is, between states 2 and 3). There is a limit to the number of in flight messages.

Limits that apply to in flight messages are unrelated to the *unlimited* number of stored messages.

For most standard queues (depending on queue traffic and message backlog), there can be a maximum of approximately 120,000 in flight messages (received from a queue by a consumer, but not yet deleted from the queue). If you reach this limit, Amazon SQS returns the `OverLimit` error message. To avoid reaching the limit, you should delete messages from the queue after they're processed. You can also increase the number of queues you use to process your messages. To request a limit increase, file a support request.

For FIFO queues, there can be a maximum of 120,000 in flight messages (received from a queue by a consumer, but not yet deleted from the queue). If you reach this limit, Amazon SQS returns no error messages.

Important

If you attempt to set the `VisibilityTimeout` to a value greater than the maximum time left, Amazon SQS returns an error. Amazon SQS doesn't automatically recalculate and increase the timeout to the maximum remaining time.
Unlike with a queue, when you change the visibility timeout for a specific message the timeout value is applied immediately but isn't saved in memory for that message.
If you don't delete a message after it is received, the visibility timeout for the message reverts to the original timeout value (not to the value you set using the `ChangeMessageVisibility` action) the next time the message is received.

Request Syntax

```
{
   "QueueUrl": "string",
   "ReceiptHandle": "string",
   "VisibilityTimeout": number
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

QueueUrl

The URL of the Amazon SQS queue whose message's visibility is changed.

Queue URLs and names are case-sensitive.

Type: String

Required: Yes

ReceiptHandle

The receipt handle associated with the message, whose visibility timeout is changed. This parameter is returned by the `ReceiveMessage` action.

Type: String

Required: Yes

VisibilityTimeout

The new value for the message's visibility timeout (in seconds). Values range: `0` to `43200`. Maximum: 12 hours.

Type: Integer

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors.

InvalidAddress

The specified ID is invalid.

HTTP Status Code: 400

InvalidSecurity

The request was not made over HTTPS or did not use SigV4 for signing.

HTTP Status Code: 400

MessageNotInflight

The specified message isn't in flight.

HTTP Status Code: 400

QueueDoesNotExist

Ensure that the `QueueUrl` is correct and that the queue has not been deleted.

HTTP Status Code: 400

ReceiptHandleIsInvalid

The specified receipt handle isn't valid.

HTTP Status Code: 400

RequestThrottled

The request was denied due to request throttling.

- Exceeds the permitted request rate for the queue or for the recipient of the request.
- Ensure that the request rate is within the Amazon SQS limits for sending messages. For more information, see Amazon SQS quotas in the *Amazon SQS Developer Guide*.

HTTP Status Code: 400

UnsupportedOperation

Error code 400. Unsupported operation.

HTTP Status Code: 400

Examples

The following example queries request changes the visibility timeout for a message to 60 seconds. The structure of AUTHPARAMS depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the *AWS General Reference*.

Example

Using AWS JSON protocol (Default)

Sample Request

```
POST / HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Target: AmazonSQS.ChangeMessageVisibility
X-Amz-Date: <Date>
Content-Type: application/x-amz-json-1.0
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
{
```

```
    "QueueUrl": "https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue/",
    "ReceiptHandle": "AQEBaZ+j5qUoOAoxlmrCQPkBm9njMWXqemmIG6shMHCO6fV20JrQYg/
AiZ8JELwLwOu5U61W+aIX5Qzu7GGofxJuvzymr4Ph53RiR0mudj4InLSgpSspYeTRDteBye5tV/txbZDdNZxsi
+qqZA9xPnmMscKQqF6pGhnGIKrnkYGl45Nl6GPIZv62LrIRb6mSqOn1fn0yqrvmWuuY3w2UzQbaYunJWGxpzZze21EOBtyw
g7G9is
+c6K9hGniddzhLkK1tHzZKjejOU4jokaiB4nmi0dF3JqLzDsQuPF0Gi8qffhEvw56nl8QCbluSJScFhJYvoagGnDbwOnd9z
NGjWJ45dqWpiW3d2Ws7q",
    "VisibilityTimeout": 60
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <requestId>
Content-Length: 0
Date: <Date>
Content-Type: application/x-amz-json-1.0
```

Example

Using AWS query protocol

Sample Request

```
POST /177715257436/MyQueue HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Date: <Date>
Content-Type: application/x-www-form-urlencoded
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
Action=ChangeMessageVisibility
&VisibilityTimeout=60
&ReceiptHandle=AQEBwPTK2fT2gy97H1iyU5in9umgT+Y4IOxyKGOzpZa8iemEqoR5/
aPn0xAodmiVTzyrW7S4e8XwcWbB04XK92jIQzUpiGwRFA4Dl7r3GOw84Qzq/0OBQe/
JaKxJw6iilafYA5fo1SJQo5Wg8xXbJHTVlJqgvTXd/
UtlByLMhWMi0JMra1UUjYiPsGtYUpLVnOaRkYSPvzRnFFYUbcqCW9lm2BijQKK6KNOZyCCfIh8TooE5i4P2L9N3o9yUHwMd
```

Sample Response

```
HTTP/1.1 200 OK
<?xml version="1.0"?>
```

```
<ChangeMessageVisibilityResponse xmlns="http://queue.amazonaws.com/doc/2012-11-05/">
    <ResponseMetadata>
        <RequestId>6a7a282a-d013-4a59-aba9-335b0fa48bed</RequestId>
    </ResponseMetadata>
</ChangeMessageVisibilityResponse>
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python (Boto3)
- AWS SDK for Ruby V3

ChangeMessageVisibilityBatch

Changes the visibility timeout of multiple messages. This is a batch version of `ChangeMessageVisibility`. The result of the action on each message is reported individually in the response. You can send up to 10 `ChangeMessageVisibility` requests with each `ChangeMessageVisibilityBatch` action.

> **Important**
>
> Because the batch request can result in a combination of successful and unsuccessful actions, you should check for batch errors even when the call returns an HTTP status code of `200`.

Request Syntax

```
{
   "Entries": [
      {
         "Id": "string",
         "ReceiptHandle": "string",
         "VisibilityTimeout": number
      }
   ],
   "QueueUrl": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

Entries

Lists the receipt handles of the messages for which the visibility timeout must be changed.

Type: Array of ChangeMessageVisibilityBatchRequestEntry objects

Required: Yes

QueueUrl

The URL of the Amazon SQS queue whose messages' visibility is changed.

Queue URLs and names are case-sensitive.

Type: String

Required: Yes

Response Syntax

```
{
   "Failed": [
      {
         "Code": "string",
         "Id": "string",
         "Message": "string",
         "SenderFault": boolean
      }
   ],
   "Successful": [
      {
         "Id": "string"
      }
   ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Failed

A list of `BatchResultErrorEntry` items.

Type: Array of BatchResultErrorEntry objects

Successful

A list of `ChangeMessageVisibilityBatchResultEntry` items.

Type: Array of ChangeMessageVisibilityBatchResultEntry objects

Errors

For information about the errors that are common to all actions, see Common Errors.

BatchEntryIdsNotDistinct

Two or more batch entries in the request have the same `Id`.

HTTP Status Code: 400

EmptyBatchRequest

The batch request doesn't contain any entries.

HTTP Status Code: 400

InvalidAddress

The specified ID is invalid.

HTTP Status Code: 400

InvalidBatchEntryId

The `Id` of a batch entry in a batch request doesn't abide by the specification.

HTTP Status Code: 400

InvalidSecurity

The request was not made over HTTPS or did not use SigV4 for signing.

HTTP Status Code: 400

QueueDoesNotExist

Ensure that the `QueueUrl` is correct and that the queue has not been deleted.

HTTP Status Code: 400

RequestThrottled

The request was denied due to request throttling.

- Exceeds the permitted request rate for the queue or for the recipient of the request.
- Ensure that the request rate is within the Amazon SQS limits for sending messages. For more information, see Amazon SQS quotas in the *Amazon SQS Developer Guide*.

HTTP Status Code: 400

TooManyEntriesInBatchRequest

The batch request contains more entries than permissible. For Amazon SQS, the maximum number of entries you can include in a single SendMessageBatch, DeleteMessageBatch, or ChangeMessageVisibilityBatch request is 10.

HTTP Status Code: 400

UnsupportedOperation

Error code 400. Unsupported operation.

HTTP Status Code: 400

Examples

`ChangeMessageVisibilityBatch` request changes the visibility timeout settings for two messages. You must URL-encode the entire URL. However, in this example only the message body is URL-encoded to make the example easier to read.

Example

Using AWS JSON protocol (Default)

Sample Request

```
POST / HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Target: AmazonSQS.ChangeMessageVisibilityBatch
X-Amz-Date: <Date>
Content-Type: application/x-amz-json-1.0
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
{
    "QueueUrl": "https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue/",
```

```
    "Entries": [
        {
            "Id": "change_visibility_msg_1",
            "ReceiptHandle": "AQEBt7JdeB+HGhLUgVF4/Gc+AvTPvgUL
+AVpoHQPb3Gw9U6bRBxx4KYy3U5bk+hBLDGAj84UxMMTG1uhCJLMuS+5MrRkuEE0pgJr6B6J12Xk
+qstDQNsRuamy21kSzjmDWhdKNexq8mcMy8dbmSYLa
+GytNMShSzyOhrBBUbcwkTdbJPW9czNs6nR4Wb8CteMPwzYCYrYHaM/
tePfqcMwlq+d7PSXTM965g2DrTP5pF5puB00BA5MkHTkZyMkUZyENTS704HH
+uZW06+7IWUTPcBma8aFt8MY0eeDnOhBtM8SD6fcnNxHXHDOmSkE6cpA68ew5wwYkPdQ2U341SxJbNWGPEfmzCtRJKs7Mc3
+cdjfyNR3QzXiVucoqS6mmRDnVe",
            "VisibilityTimeout": 45
        },
        {
            "Id": "change_visibility_msg_2",
            "ReceiptHandle": "AQEB3LQoW7GQWgodQCEJXHjMvO/QkeCHiRldRfLC/E6RUggm
+BjpthqxfoUOUn6Vs271qmrBaufFqEmnMKgk2n1EuUBne1pe+hZcrDE8IveUUPmqkUT54FGhAAjPX3oEIryz/
XeQ/muKAuLclcZvt2Q+ZDPW8DvZqMa1RoHxOqSq+6kQ4PwgQxB+VqDYvIc/LpHOoL4PTROBXgLPjWrzz/
knK6HTzKpqC4ESvFdJ/dkk2nvS0iqYOly5VQknK/lv/rTUOgEYevjJSrNLIPDgZGyvgcLwbm6+yo1cW/
c9cPDiVm96gIhVkuiCZ1gtskoOtyroZVPcY71clDG2EPZJeY8akMd3u+sXEMWhiOPFs1cgWQs2ugsL
+vdwMCbsZRkXbJv7",
            "VisibilityTimeout": 45
        }
    ]
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <requestId>
Content-Length: <PayloadSizeBytes>
Date: <Date>
Content-Type: application/x-amz-json-1.0
{
    "Failed": [],
    "Successful": [
        {
            "Id": "change_visibility_msg_1"
        },
        {
            "Id": "change_visibility_msg_2"
        }
    ]
}
```

Example

Using AWS query protocol

Sample Request

```
POST /177715257436/MyQueue HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Date: <Date>
Content-Type: application/x-www-form-urlencoded
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
Action=ChangeMessageVisibilityBatch
&ChangeMessageVisibilityBatchRequestEntry.1.Id=change_visibility_msg_2
&ChangeMessageVisibilityBatchRequestEntry.1.ReceiptHandle=gfk0T0R0waama4fVFffkjKzmhMCymjQvfTFk2
%2F%2Fa1IXW6AqaWhGsEPaLm3Vf6IiWqdM8u5imB%2BNTwj3tQRzOWdTOePjOjPcTpRxBtXix
%2BEvwJOZUma9wabv%2BSw6ZHjwmNcVDx8dZXJhVp16Bksiox
%2FGrUvrVTCJRTWTLc59oHLLF8sEkKzRmGNzTDGTiV
%2BYjHfQj60FD3rVaXmzTsoNxRhKJ72uIHVMGVQiAGgBX6HGv 9LDmYhPXw4hy%2FNgIg%3D%3D
&ChangeMessageVisibilityBatchRequestEntry.1.VisibilityTimeout=45
&ChangeMessageVisibilityBatchRequestEntry.2.Id=change_visibility_msg_3
&ChangeMessageVisibilityBatchRequestEntry.2.ReceiptHandle=gfk0T0R0waama4f
 VFffkjKzmhMCymjQvfTFk2LxT33FUgBz3%2BnougdeLKWSscPU1%2FXgx%2BxcNnjnQQ3U30q
 OumIE6AdAv3w%2F%2Fa1IXW6AqaWhGsEPaLm3Vf6IiWqdM8u5imB%2BNTwj3tQRzOWdTOePjO
 sogjZM%2F7kzn4Ew27XLU9I%2FYaWYmKvDbq%2Fk3HKVB9HfB43kE49atP2aWrzNL4yunG41Q
 4cfRRtfJdcGQGNHQ2%2Byd0Usf5qR1dZr1iDo5xk946eQat83AxTRP%2BY4Qi0V7FAeSLH9su
 9xpX6HGv9LDmYhPXw4hy%2FNgIg%3D%3D
&ChangeMessageVisibilityBatchRequestEntry.2.VisibilityTimeout=45
```

Sample Response

```
HTTP/1.1 200 OK
<?xml version="1.0"?>
<ChangeMessageVisibilityBatchResponse xmlns="http://queue.amazonaws.com/
doc/2012-11-05/">
    <ChangeMessageVisibilityBatchResult>
        <ChangeMessageVisibilityBatchResultEntry>
            <Id>change_visibility_msg_2</Id>
        </ChangeMessageVisibilityBatchResultEntry>
        <ChangeMessageVisibilityBatchResultEntry>
            <Id>change_visibility_msg_3</Id>
        </ChangeMessageVisibilityBatchResultEntry>
```

```
    </ChangeMessageVisibilityBatchResult>
    <ResponseMetadata>
        <RequestId>ca9668f7-ab1b-4f7a-8859-f15747ab17a7</RequestId>
    </ResponseMetadata>
</ChangeMessageVisibilityBatchResponse>
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python (Boto3)
- AWS SDK for Ruby V3

CreateQueue

Creates a new standard or FIFO queue. You can pass one or more attributes in the request. Keep the following in mind:

- If you don't specify the `FifoQueue` attribute, Amazon SQS creates a standard queue.

  > **Note**
  >
  > You can't change the queue type after you create it and you can't convert an existing standard queue into a FIFO queue. You must either create a new FIFO queue for your application or delete your existing standard queue and recreate it as a FIFO queue. For more information, see Moving From a standard queue to a FIFO queue in the *Amazon SQS Developer Guide*.

- If you don't provide a value for an attribute, the queue is created with the default value for the attribute.
- If you delete a queue, you must wait at least 60 seconds before creating a queue with the same name.

To successfully create a new queue, you must provide a queue name that adheres to the limits related to queues and is unique within the scope of your queues.

> **Note**
>
> After you create a queue, you must wait at least one second after the queue is created to be able to use the queue.

To retrieve the URL of a queue, use the `GetQueueUrl` action. This action only requires the `QueueName` parameter.

When creating queues, keep the following points in mind:

- If you specify the name of an existing queue and provide the exact same names and values for all its attributes, the `CreateQueue` action will return the URL of the existing queue instead of creating a new one.

- If you attempt to create a queue with a name that already exists but with different attribute names or values, the `CreateQueue` action will return an error. This ensures that existing queues are not inadvertently altered.

> **Note**
>
> Cross-account permissions don't apply to this action. For more information, see Grant cross-account permissions to a role and a username in the *Amazon SQS Developer Guide*.

Request Syntax

```
{
   "Attributes": {
      "string" : "string"
   },
   "QueueName": "string",
   "tags": {
      "string" : "string"
   }
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

Attributes

A map of attributes with their corresponding values.

The following lists the names, descriptions, and values of the special request parameters that the `CreateQueue` action uses:

- `DelaySeconds` – The length of time, in seconds, for which the delivery of all messages in the queue is delayed. Valid values: An integer from 0 to 900 seconds (15 minutes). Default: 0.
- `MaximumMessageSize` – The limit of how many bytes a message can contain before Amazon SQS rejects it. Valid values: An integer from 1,024 bytes (1 KiB) to 1,048,576 bytes (1 MiB). Default: 1,048,576 bytes (1 MiB).

- `MessageRetentionPeriod` – The length of time, in seconds, for which Amazon SQS retains a message. Valid values: An integer from 60 seconds (1 minute) to 1,209,600 seconds (14 days). Default: 345,600 (4 days). When you change a queue's attributes, the change can take up to 60 seconds for most of the attributes to propagate throughout the Amazon SQS system. Changes made to the `MessageRetentionPeriod` attribute can take up to 15 minutes and will impact existing messages in the queue potentially causing them to be expired and deleted if the `MessageRetentionPeriod` is reduced below the age of existing messages.
- `Policy` – The queue's policy. A valid AWS policy. For more information about policy structure, see Overview of AWS IAM Policies in the *IAM User Guide*.
- `ReceiveMessageWaitTimeSeconds` – The length of time, in seconds, for which a `ReceiveMessage` action waits for a message to arrive. Valid values: An integer from 0 to 20 (seconds). Default: 0.
- `VisibilityTimeout` – The visibility timeout for the queue, in seconds. Valid values: An integer from 0 to 43,200 (12 hours). Default: 30. For more information about the visibility timeout, see Visibility Timeout in the *Amazon SQS Developer Guide*.

The following attributes apply only to dead-letter queues:

- `RedrivePolicy` – The string that includes the parameters for the dead-letter queue functionality of the source queue as a JSON object. The parameters are as follows:
  - `deadLetterTargetArn` – The Amazon Resource Name (ARN) of the dead-letter queue to which Amazon SQS moves messages after the value of `maxReceiveCount` is exceeded.
  - `maxReceiveCount` – The number of times a message is delivered to the source queue before being moved to the dead-letter queue. Default: 10. When the `ReceiveCount` for a message exceeds the `maxReceiveCount` for a queue, Amazon SQS moves the message to the dead-letter-queue.
- `RedriveAllowPolicy` – The string that includes the parameters for the permissions for the dead-letter queue redrive permission and which source queues can specify dead-letter queues as a JSON object. The parameters are as follows:
  - `redrivePermission` – The permission type that defines which source queues can specify the current queue as the dead-letter queue. Valid values are:
    - `allowAll` – (Default) Any source queues in this AWS account in the same Region can specify this queue as the dead-letter queue.
    - `denyAll` – No source queues can specify this queue as the dead-letter queue.

  - byQueue – Only queues specified by the `sourceQueueArns` parameter can specify this queue as the dead-letter queue.
- `sourceQueueArns` – The Amazon Resource Names (ARN)s of the source queues that can specify this queue as the dead-letter queue and redrive messages. You can specify this parameter only when the `redrivePermission` parameter is set to `byQueue`. You can specify up to 10 source queue ARNs. To allow more than 10 source queues to specify dead-letter queues, set the `redrivePermission` parameter to `allowAll`.

> **Note**
>
> The dead-letter queue of a FIFO queue must also be a FIFO queue. Similarly, the dead-letter queue of a standard queue must also be a standard queue.

The following attributes apply only to server-side-encryption:

- `KmsMasterKeyId` – The ID of an AWS managed customer master key (CMK) for Amazon SQS or a custom CMK. For more information, see Key Terms. While the alias of the AWS managed CMK for Amazon SQS is always `alias/aws/sqs`, the alias of a custom CMK can, for example, be `alias/MyAlias` . For more examples, see KeyId in the *AWS Key Management Service API Reference*.
- `KmsDataKeyReusePeriodSeconds` – The length of time, in seconds, for which Amazon SQS can reuse a data key to encrypt or decrypt messages before calling AWS KMS again. An integer representing seconds, between 60 seconds (1 minute) and 86,400 seconds (24 hours). Default: 300 (5 minutes). A shorter time period provides better security but results in more calls to KMS which might incur charges after Free Tier. For more information, see How Does the Data Key Reuse Period Work?
- `SqsManagedSseEnabled` – Enables server-side queue encryption using SQS owned encryption keys. Only one server-side encryption option is supported per queue (for example, SSE-KMS or SSE-SQS).

The following attributes apply only to FIFO (first-in-first-out) queues:

- `FifoQueue` – Designates a queue as FIFO. Valid values are `true` and `false`. If you don't specify the `FifoQueue` attribute, Amazon SQS creates a standard queue. You can provide this attribute only during queue creation. You can't change it for an existing queue. When you set this attribute, you must also provide the `MessageGroupId` for your messages explicitly.

  For more information, see FIFO queue logic in the *Amazon SQS Developer Guide*.

- `ContentBasedDeduplication` – Enables content-based deduplication. Valid values are `true` and `false`. For more information, see Exactly-once processing in the *Amazon SQS Developer Guide*. Note the following:
  - Every message must have a unique `MessageDeduplicationId`.
    - You may provide a `MessageDeduplicationId` explicitly.
    - If you aren't able to provide a `MessageDeduplicationId` and you enable `ContentBasedDeduplication` for your queue, Amazon SQS uses a SHA-256 hash to generate the `MessageDeduplicationId` using the body of the message (but not the attributes of the message).
    - If you don't provide a `MessageDeduplicationId` and the queue doesn't have `ContentBasedDeduplication` set, the action fails with an error.
    - If the queue has `ContentBasedDeduplication` set, your `MessageDeduplicationId` overrides the generated one.
  - When `ContentBasedDeduplication` is in effect, messages with identical content sent within the deduplication interval are treated as duplicates and only one copy of the message is delivered.
  - If you send one message with `ContentBasedDeduplication` enabled and then another message with a `MessageDeduplicationId` that is the same as the one generated for the first `MessageDeduplicationId`, the two messages are treated as duplicates and only one copy of the message is delivered.

The following attributes apply only to high throughput for FIFO queues:

- `DeduplicationScope` – Specifies whether message deduplication occurs at the message group or queue level. Valid values are `messageGroup` and `queue`.
- `FifoThroughputLimit` – Specifies whether the FIFO queue throughput quota applies to the entire queue or per message group. Valid values are `perQueue` and `perMessageGroupId`. The `perMessageGroupId` value is allowed only when the value for `DeduplicationScope` is `messageGroup`.

To enable high throughput for FIFO queues, do the following:

- Set `DeduplicationScope` to `messageGroup`.
- Set `FifoThroughputLimit` to `perMessageGroupId`.

If you set these attributes to anything other than the values shown for enabling high throughput, normal throughput is in effect and deduplication occurs as specified.

For information on throughput quotas, see Quotas related to messages in the *Amazon SQS Developer Guide*.

Type: String to string map

Valid Keys: `All | Policy | VisibilityTimeout | MaximumMessageSize | MessageRetentionPeriod | ApproximateNumberOfMessages | ApproximateNumberOfMessagesNotVisible | CreatedTimestamp | LastModifiedTimestamp | QueueArn | ApproximateNumberOfMessagesDelayed | DelaySeconds | ReceiveMessageWaitTimeSeconds | RedrivePolicy | FifoQueue | ContentBasedDeduplication | KmsMasterKeyId | KmsDataKeyReusePeriodSeconds | DeduplicationScope | FifoThroughputLimit | RedriveAllowPolicy | SqsManagedSseEnabled`

Required: No

QueueName

The name of the new queue. The following limits apply to this name:

- A queue name can have up to 80 characters.
- Valid values: alphanumeric characters, hyphens (-), and underscores (_).
- A FIFO queue name must end with the `.fifo` suffix.

Queue URLs and names are case-sensitive.

Type: String

Required: Yes

tags

Add cost allocation tags to the specified Amazon SQS queue. For an overview, see Tagging Your Amazon SQS Queues in the *Amazon SQS Developer Guide*.

When you use queue tags, keep the following guidelines in mind:

- Adding more than 50 tags to a queue isn't recommended.
- Tags don't have any semantic meaning. Amazon SQS interprets tags as character strings.
- Tags are case-sensitive.
- A new tag with a key identical to that of an existing tag overwrites the existing tag.

For a full list of tag restrictions, see Quotas related to queues in the *Amazon SQS Developer Guide*.

> **Note**
>
> To be able to tag a queue on creation, you must have the `sqs:CreateQueue` and `sqs:TagQueue` permissions.
> Cross-account permissions don't apply to this action. For more information, see Grant cross-account permissions to a role and a username in the *Amazon SQS Developer Guide*.

Type: String to string map

Required: No

Response Syntax

```
{
   "QueueUrl": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

QueueUrl

The URL of the created Amazon SQS queue.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

InvalidAddress

The specified ID is invalid.

HTTP Status Code: 400

InvalidAttributeName

The specified attribute doesn't exist.

HTTP Status Code: 400

InvalidAttributeValue

A queue attribute value is invalid.

HTTP Status Code: 400

InvalidSecurity

The request was not made over HTTPS or did not use SigV4 for signing.

HTTP Status Code: 400

QueueDeletedRecently

You must wait 60 seconds after deleting a queue before you can create another queue with the same name.

HTTP Status Code: 400

QueueNameExists

A queue with this name already exists. Amazon SQS returns this error only if the request includes attributes whose values differ from those of the existing queue.

HTTP Status Code: 400

RequestThrottled

The request was denied due to request throttling.

- Exceeds the permitted request rate for the queue or for the recipient of the request.
- Ensure that the request rate is within the Amazon SQS limits for sending messages. For more information, see Amazon SQS quotas in the *Amazon SQS Developer Guide*.

HTTP Status Code: 400

UnsupportedOperation

Error code 400. Unsupported operation.

HTTP Status Code: 400

Examples

The following example query requests create a new queue named MyQueue. The structure of AUTHPARAMS depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the *AWS General Reference*.

Example

Using AWS JSON protocol (Default)

Sample Request

```
POST / HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Target: AmazonSQS.CreateQueue
X-Amz-Date: <Date>
Content-Type: application/x-amz-json-1.0
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
{
    "QueueName":"MyQueue",
    "Attributes": {
        "VisibilityTimeout": "40"
    },
    "tags": {
        "QueueType": "Production"
    }
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <requestId>
Content-Length: <PayloadSizeBytes>
Date: <Date>
Content-Type: application/x-amz-json-1.0
{
    "QueueUrl":"https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue"
```

```
}
```

Example

Using AWS query protocol

Sample Request

```
POST / HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
Content-Type: application/x-www-form-urlencoded
X-Amz-Date: <Date>
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
Action=CreateQueue
&QueueName=MyQueue
&Attribute.1.Name=VisibilityTimeout
&Attribute.1.Value=40
&Tag.Key=QueueType
&Tag.Value=Production
```

Sample Response

```
HTTP/1.1 200 OK
<?xml version="1.0"?>
<CreateQueueResponse xmlns="http://queue.amazonaws.com/doc/2012-11-05/">
    <CreateQueueResult>
        <QueueUrl>https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue</QueueUrl>
    </CreateQueueResult>
    <ResponseMetadata>
        <RequestId>9b20926c-8b35-5d8e-9559-ce1c22e754dc</RequestId>
    </ResponseMetadata>
</CreateQueueResponse>
```

Example

The following example creates a delay queue which hides each message from consumers for the first 45 seconds that the message is in the queue by calling the `CreateQueue` action with the `DelaySeconds` attribute set to 45 seconds.

> **Note**
>
> Queue URLs and names are case-sensitive.

Sample Request

```
https://sqs.us-east-2.amazonaws.com/123456789012/MyQueue/
?Action=CreateQueue
&QueueName=MyQueue
&Attribute.1.Name=DelaySeconds
&Attribute.1.Value=45
&Expires=2020-12-20T22%3A52%3A43PST
&Version=2012-11-05
&AUTHPARAMS
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python (Boto3)
- AWS SDK for Ruby V3

DeleteMessage

Deletes the specified message from the specified queue. To select the message to delete, use the `ReceiptHandle` of the message (*not* the `MessageId` which you receive when you send the message). Amazon SQS can delete a message from a queue even if a visibility timeout setting causes the message to be locked by another consumer. Amazon SQS automatically deletes messages left in a queue longer than the retention period configured for the queue.

> **Note**
>
> Each time you receive a message, meaning when a consumer retrieves a message from the queue, it comes with a unique `ReceiptHandle`. If you receive the same message more than once, you will get a different `ReceiptHandle` each time. When you want to delete a message using the `DeleteMessage` action, you must use the `ReceiptHandle` from the most recent time you received the message. If you use an old `ReceiptHandle`, the request will succeed, but the message might not be deleted.
> For standard queues, it is possible to receive a message even after you delete it. This might happen on rare occasions if one of the servers which stores a copy of the message is unavailable when you send the request to delete the message. The copy remains on the server and might be returned to you during a subsequent receive request. You should ensure that your application is idempotent, so that receiving a message more than once does not cause issues.

Request Syntax

```
{
   "QueueUrl": "string",
   "ReceiptHandle": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

QueueUrl

The URL of the Amazon SQS queue from which messages are deleted.

Queue URLs and names are case-sensitive.

Type: String

Required: Yes

ReceiptHandle

The receipt handle associated with the message to delete.

Type: String

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors.

InvalidAddress

The specified ID is invalid.

HTTP Status Code: 400

InvalidIdFormat

This error has been deprecated.

The specified receipt handle isn't valid for the current version.

HTTP Status Code: 400

InvalidSecurity

The request was not made over HTTPS or did not use SigV4 for signing.

HTTP Status Code: 400

QueueDoesNotExist

Ensure that the `QueueUrl` is correct and that the queue has not been deleted.

HTTP Status Code: 400

ReceiptHandleIsInvalid

The specified receipt handle isn't valid.

HTTP Status Code: 400

RequestThrottled

The request was denied due to request throttling.

- Exceeds the permitted request rate for the queue or for the recipient of the request.
- Ensure that the request rate is within the Amazon SQS limits for sending messages. For more information, see Amazon SQS quotas in the *Amazon SQS Developer Guide*.

HTTP Status Code: 400

UnsupportedOperation

Error code 400. Unsupported operation.

HTTP Status Code: 400

Examples

The following example query request deletes a message from the queue named `MyQueue`. The structure of `AUTHPARAMS` depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the *AWS General Reference*.

Example

Using AWS JSON protocol (Default)

Sample Request

```
POST / HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
```

```
X-Amz-Target: AmazonSQS.DeleteMessage
X-Amz-Date: <Date>
Content-Type: application/x-amz-json-1.0
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
{
    "QueueUrl": "https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue/",
    "ReceiptHandle": "AQEB3LQoW7GQWgodQCEJXHjMvO/QkeCHiRldRfLC/E6RUggm
+BjpthqxfoUOUn6Vs271qmrBaufFqEmnMKgk2n1EuUBne1pe+hZcrDE8IveUUPmqkUT54FGhAAjPX3oEIryz/
XeQ/muKAuLclcZvt2Q+ZDPW8DvZqMa1RoHxOqSq+6kQ4PwgQxB+VqDYvIc/LpHOoL4PTROBXgLPjWrzz/
knK6HTzKpqC4ESvFdJ/dkk2nvS0iqYOly5VQknK/lv/rTUOgEYevjJSrNLIPDgZGyvgcLwbm6+yo1cW/
c9cPDiVm96gIhVkuiCZ1gtskoOtyroZVPcY71clDG2EPZJeY8akMd3u+sXEMWhiOPFs1cgWQs2ugsL
+vdwMCbsZRkXbJv7"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <requestId>
Content-Length: 0
Date: <Date>
Content-Type: application/x-amz-json-1.0
```

Example

Using AWS query protocol

Sample Request

```
POST /177715257436/MyQueue/ HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
Content-Type: application/x-www-form-urlencoded
X-Amz-Date: <Date>
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
Action=DeleteMessage
&ReceiptHandle=AQEBMeG2RcZZrIcgBkDFb6lHqL9B9tbbEHNh+4uxMIG
+CPupPjqJtRswDlOr6hOTzgcq105i0iZNci5GS5RTnHTkD2zipM9gHfSP2tWPhY7HHsU5GCTZ
+egzS5HiEvmGZ71g71Lucdk7mes1/
WGXnmU27K26Koo9GGrB0AKTv16dync1ezCMNyrBHEMUyIWS2lUTbrSj7fw93dgZSg2eWTk+thSVUB/ibOwpmj
+wBN99nKQQklsZHtZd4exT1V3JHwP4kqz
```

```
+D3C2RGn7js3nNdFpH41lBH8rCTZDU8DQp9eQNHLIL6RUf1WrI8gv8L7NErGlIH4Y3wZbFEOMKilVHenfpP2G6ElMuxyM3y
+qdlZq4m00VGIIZeMg9PPmVsLtB7u9mruLyNFraN5ihKMjzQoKgA==
```

Sample Response

```
HTTP/1.1 200 OK
<?xml version="1.0"?>
<DeleteMessageResponse xmlns="http://queue.amazonaws.com/doc/2012-11-05/">
    <ResponseMetadata>
        <RequestId>b5293cb5-d306-4a17-9048-b263635abe42</RequestId>
    </ResponseMetadata>
</DeleteMessageResponse>
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python (Boto3)
- AWS SDK for Ruby V3

DeleteMessageBatch

Deletes up to ten messages from the specified queue. This is a batch version of `DeleteMessage`. The result of the action on each message is reported individually in the response.

> **Important**
>
> Because the batch request can result in a combination of successful and unsuccessful actions, you should check for batch errors even when the call returns an HTTP status code of `200`.

Request Syntax

```
{
   "Entries": [
      {
         "Id": "string",
         "ReceiptHandle": "string"
      }
   ],
   "QueueUrl": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

Entries

Lists the receipt handles for the messages to be deleted.

Type: Array of DeleteMessageBatchRequestEntry objects

Required: Yes

QueueUrl

The URL of the Amazon SQS queue from which messages are deleted.

Queue URLs and names are case-sensitive.

Type: String

Required: Yes

Response Syntax

```
{
   "Failed": [
      {
         "Code": "string",
         "Id": "string",
         "Message": "string",
         "SenderFault": boolean
      }
   ],
   "Successful": [
      {
         "Id": "string"
      }
   ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Failed

A list of `BatchResultErrorEntry` items.

Type: Array of BatchResultErrorEntry objects

Successful

A list of `DeleteMessageBatchResultEntry` items.

Type: Array of DeleteMessageBatchResultEntry objects

Errors

For information about the errors that are common to all actions, see Common Errors.

BatchEntryIdsNotDistinct

Two or more batch entries in the request have the same `Id`.

HTTP Status Code: 400

EmptyBatchRequest

The batch request doesn't contain any entries.

HTTP Status Code: 400

InvalidAddress

The specified ID is invalid.

HTTP Status Code: 400

InvalidBatchEntryId

The `Id` of a batch entry in a batch request doesn't abide by the specification.

HTTP Status Code: 400

InvalidSecurity

The request was not made over HTTPS or did not use SigV4 for signing.

HTTP Status Code: 400

QueueDoesNotExist

Ensure that the `QueueUrl` is correct and that the queue has not been deleted.

HTTP Status Code: 400

RequestThrottled

The request was denied due to request throttling.

- Exceeds the permitted request rate for the queue or for the recipient of the request.
- Ensure that the request rate is within the Amazon SQS limits for sending messages. For more information, see Amazon SQS quotas in the *Amazon SQS Developer Guide*.

HTTP Status Code: 400

TooManyEntriesInBatchRequest

The batch request contains more entries than permissible. For Amazon SQS, the maximum number of entries you can include in a single SendMessageBatch, DeleteMessageBatch, or ChangeMessageVisibilityBatch request is 10.

HTTP Status Code: 400

UnsupportedOperation

Error code 400. Unsupported operation.

HTTP Status Code: 400

Examples

The following examples show how a `DeleteMessageBatch` request deletes two messages. You must URL-encode the entire URL. However, in this example only the message body is URL-encoded to make the example easier to read. The structure of `AUTHPARAMS` depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the *AWS General Reference*.

Example

Using AWS JSON protocol (Default)

Sample Request

```
POST / HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Target: AmazonSQS.DeleteMessageBatch
X-Amz-Date: <Date>
Content-Type: application/x-amz-json-1.0
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
{
    "QueueUrl": "https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue/",
    "Entries": [
        {
            "Id": "msg1",
            "ReceiptHandle": "AQEBaZ+j5qUoOAoxlmrCQPkBm9njMWXqemmIG6shMHCO6fV20JrQYg/
AiZ8JELwLwOu5U61W+aIX5Qzu7GGofxJuvzymr4Ph53RiR0mudj4InLSgpSspYeTRDteBye5tV/txbZDdNZxsi
```

```
+qqZA9xPnmMscKQqF6pGhnGIKrnkYGl45Nl6GPIZv62LrIRb6mSqOn1fn0yqrvmWuuY3w2UzQbaYunJWGxpzZze21EOBtyw
g7G9is
+c6K9hGniddzhLkK1tHzZKjejOU4jokaiB4nmi0dF3JqLzDsQuPF0Gi8qffhEvw56nl8QCbluSJScFhJYvoagGnDbwOnd9z
NGjWJ45dqWpiW3d2Ws7q"
        },
        {
            "Id": "msg2",
            "ReceiptHandle": "AQEB3LQoW7GQWgodQCEJXHjMvO/QkeCHiRldRfLC/E6RUggm
+BjpthqxfoUOUn6Vs271qmrBaufFqEmnMKgk2n1EuUBne1pe+hZcrDE8IveUUPmqkUT54FGhAAjPX3oEIryz/
XeQ/muKAuLclcZvt2Q+ZDPW8DvZqMa1RoHxOqSq+6kQ4PwgQxB+VqDYvIc/LpHOoL4PTROBXgLPjWrzz/
knK6HTzKpqC4ESvFdJ/dkk2nvS0iqYOly5VQknK/lv/rTUOgEYevjJSrNLIPDgZGyvgcLwbm6+yo1cW/
c9cPDiVm96gIhVkuiCZ1gtskoOtyroZVPcY71clDG2EPZJeY8akMd3u+sXEMWhiOPFs1cgWQs2ugsL
+vdwMCbsZRkXbJv7"
        }
    ]
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <requestId>
Content-Length: <>
Date: <Date>
Content-Type: application/x-amz-json-1.0
{
    "Failed": [],
    "Successful": [
        {
            "Id": "msg2"
        },
        {
            "Id": "msg1"
        }
    ]
}
```

Example

Using AWS query protocol

Sample Request

```
POST /177715257436/MyQueue/ HTTP/1.1
```

```
Host: sqs.us-east-1.amazonaws.com
Content-Type: application/x-www-form-urlencoded
X-Amz-Date: <Date>
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
Action=DeleteMessageBatch
&DeleteMessageBatchRequestEntry.1.Id=msg1
&DeleteMessageBatchRequestEntry.1.ReceiptHandle=gfk0T0R0waama4fVFffkjPQrrvzMrOg0fTFk2LxT33EuB8w
%2F5LeGPpYTLzjqLQxyQYaQALeSNHb0us3uE84uujxpBhsDkZUQkjFFkNqBXn48xlMcVhTcI3YLH%2Bd
%2BIqetIOHgBCZAPx6r%2B09dWaBXei6nbK5Ygih21DCDdAwFV68Jo8DXhb3ErEfoDqx7vyvC5nCpdwqv%2BJhU
%2FTNGjNN8t51v5c%2FAXvQsAzyZVNapxUrHIt4NxRhKJ72uICcxruyE8eRXlxIVNgeNP8ZEDcw7zZU1Zw%3D
%3D
&DeleteMessageBatchRequestEntry.2.Id=msg2
&DeleteMessageBatchRequestEntry.2.ReceiptHandle=gfk0T0R0waama4fVFffkjKzmhMCymjQvfTFk2LxT33G4ms5
%2F%2Fa1IXW6AqaWhGsEPaLm3Vf6IiWqdM8u5imB%2BNTwj3tQRzOWdTOePjOjPcTpRxBtXix
%2BEvwJOZUma9wabv%2BSw6ZHjwmNcVDx8dZXJhVp16Bksiox
%2FGrUvrVTCJRTWTLc59oHLLF8sEkKzRmGNzTDGTiV%2BYjHfQj60FD3rVaXmzTsoNxRhKJ72uIHVMGVQiAGgB
%2BqAbSqfKHDQtVOmJJgkHug%3D%3D
```

Sample Response

```
HTTP/1.1 200 OK
<?xml version="1.0"?>
<DeleteMessageBatchResponse xmlns="http://queue.amazonaws.com/doc/2012-11-05/">
    <DeleteMessageBatchResult>
        <DeleteMessageBatchResultEntry>
            <Id>msg1</Id>
        </DeleteMessageBatchResultEntry>
        <DeleteMessageBatchResultEntry>
            <Id>msg2</Id>
        </DeleteMessageBatchResultEntry>
    </DeleteMessageBatchResult>
    <ResponseMetadata>
        <RequestId>d6f86b7a-74d1-4439-b43f-196a1e29cd85</RequestId>
    </ResponseMetadata>
</DeleteMessageBatchResponse>
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python (Boto3)
- AWS SDK for Ruby V3

DeleteQueue

Deletes the queue specified by the `QueueUrl`, regardless of the queue's contents.

> **Important**
>
> Be careful with the `DeleteQueue` action: When you delete a queue, any messages in the queue are no longer available.

When you delete a queue, the deletion process takes up to 60 seconds. Requests you send involving that queue during the 60 seconds might succeed. For example, a `SendMessage` request might succeed, but after 60 seconds the queue and the message you sent no longer exist.

When you delete a queue, you must wait at least 60 seconds before creating a queue with the same name.

> **Note**
>
> Cross-account permissions don't apply to this action. For more information, see Grant cross-account permissions to a role and a username in the *Amazon SQS Developer Guide*.
> The delete operation uses the HTTP `GET` verb.

Request Syntax

```
{
   "QueueUrl": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

QueueUrl

The URL of the Amazon SQS queue to delete.

Queue URLs and names are case-sensitive.

Type: String

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors.

InvalidAddress

The specified ID is invalid.

HTTP Status Code: 400

InvalidSecurity

The request was not made over HTTPS or did not use SigV4 for signing.

HTTP Status Code: 400

QueueDoesNotExist

Ensure that the `QueueUrl` is correct and that the queue has not been deleted.

HTTP Status Code: 400

RequestThrottled

The request was denied due to request throttling.

- Exceeds the permitted request rate for the queue or for the recipient of the request.
- Ensure that the request rate is within the Amazon SQS limits for sending messages. For more information, see Amazon SQS quotas in the *Amazon SQS Developer Guide*.

HTTP Status Code: 400

UnsupportedOperation

Error code 400. Unsupported operation.

HTTP Status Code: 400

Examples

The following example query request deletes the specified queue. The structure of AUTHPARAMS depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the *AWS General Reference*.

Example

Using AWS JSON protocol (Default)

Sample Request

```
POST / HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Target: AmazonSQS.DeleteQueue
X-Amz-Date: <Date>
Content-Type: application/x-amz-json-1.0
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
{
    "QueueUrl":"https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <requestId>
Content-Length: 0
Date: <Date>
Content-Type: application/x-amz-json-1.0
```

Example

Using AWS query protocol

Sample Request

```
POST /177715257436/MyQueue/ HTTP/1.1
```

```
Host: sqs.us-east-1.amazonaws.com
Content-Type: application/x-www-form-urlencoded
X-Amz-Date: <Date>
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
Action=DeleteQueue
```

Sample Response

```
HTTP/1.1 200 OK
<?xml version="1.0"?>
<DeleteQueueResponse xmlns="http://queue.amazonaws.com/doc/2012-11-05/">
    <ResponseMetadata>
        <RequestId>22b441f1-c3c3-54e4-9f57-1d817c509a21</RequestId>
    </ResponseMetadata>
</DeleteQueueResponse>
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python (Boto3)
- AWS SDK for Ruby V3

GetQueueAttributes

Gets attributes for the specified queue.

> **Note**
>
> To determine whether a queue is FIFO, you can check whether `QueueName` ends with the `.fifo` suffix.

Request Syntax

```
{
   "AttributeNames": [ "string" ],
   "QueueUrl": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

AttributeNames

A list of attributes for which to retrieve information.

The `AttributeNames` parameter is optional, but if you don't specify values for this parameter, the request returns empty results.

> **Note**
>
> In the future, new attributes might be added. If you write code that calls this action, we recommend that you structure your code so that it can handle new attributes gracefully.

The following attributes are supported:

> **Important**
>
> The `ApproximateNumberOfMessagesDelayed`, `ApproximateNumberOfMessagesNotVisible`, and `ApproximateNumberOfMessages` metrics may not achieve consistency until at least 1 minute after the producers stop sending messages. This period is required for the queue metadata to reach eventual consistency.

- `All` – Returns all values.
- `ApproximateNumberOfMessages` – Returns the approximate number of messages available for retrieval from the queue.
- `ApproximateNumberOfMessagesDelayed` – Returns the approximate number of messages in the queue that are delayed and not available for reading immediately. This can happen when the queue is configured as a delay queue or when a message has been sent with a delay parameter.
- `ApproximateNumberOfMessagesNotVisible` – Returns the approximate number of messages that are in flight. Messages are considered to be *in flight* if they have been sent to a client but have not yet been deleted or have not yet reached the end of their visibility window.
- `CreatedTimestamp` – Returns the time when the queue was created in seconds (epoch time).
- `DelaySeconds` – Returns the default delay on the queue in seconds.
- `LastModifiedTimestamp` – Returns the time when the queue was last changed in seconds (epoch time).
- `MaximumMessageSize` – Returns the limit of how many bytes a message can contain before Amazon SQS rejects it.
- `MessageRetentionPeriod` – Returns the length of time, in seconds, for which Amazon SQS retains a message. When you change a queue's attributes, the change can take up to 60 seconds for most of the attributes to propagate throughout the Amazon SQS system. Changes made to the `MessageRetentionPeriod` attribute can take up to 15 minutes and will impact existing messages in the queue potentially causing them to be expired and deleted if the `MessageRetentionPeriod` is reduced below the age of existing messages.
- `Policy` – Returns the policy of the queue.

- `QueueArn` – Returns the Amazon resource name (ARN) of the queue.
- `ReceiveMessageWaitTimeSeconds` – Returns the length of time, in seconds, for which the `ReceiveMessage` action waits for a message to arrive.
- `VisibilityTimeout` – Returns the visibility timeout for the queue. For more information about the visibility timeout, see Visibility Timeout in the *Amazon SQS Developer Guide*.

The following attributes apply only to dead-letter queues:

- `RedrivePolicy` – The string that includes the parameters for the dead-letter queue functionality of the source queue as a JSON object. The parameters are as follows:
  - `deadLetterTargetArn` – The Amazon Resource Name (ARN) of the dead-letter queue to which Amazon SQS moves messages after the value of `maxReceiveCount` is exceeded.
  - `maxReceiveCount` – The number of times a message is delivered to the source queue before being moved to the dead-letter queue. Default: 10. When the `ReceiveCount` for a message exceeds the `maxReceiveCount` for a queue, Amazon SQS moves the message to the dead-letter-queue.
- `RedriveAllowPolicy` – The string that includes the parameters for the permissions for the dead-letter queue redrive permission and which source queues can specify dead-letter queues as a JSON object. The parameters are as follows:
  - `redrivePermission` – The permission type that defines which source queues can specify the current queue as the dead-letter queue. Valid values are:
    - `allowAll` – (Default) Any source queues in this AWS account in the same Region can specify this queue as the dead-letter queue.
    - `denyAll` – No source queues can specify this queue as the dead-letter queue.
    - `byQueue` – Only queues specified by the `sourceQueueArns` parameter can specify this queue as the dead-letter queue.
  - `sourceQueueArns` – The Amazon Resource Names (ARN)s of the source queues that can specify this queue as the dead-letter queue and redrive messages. You can specify this parameter only when the `redrivePermission` parameter is set to `byQueue`. You can specify up to 10 source queue ARNs. To allow more than 10 source queues to specify dead-letter queues, set the `redrivePermission` parameter to `allowAll`.

> **Note**
>
> The dead-letter queue of a FIFO queue must also be a FIFO queue. Similarly, the dead-letter queue of a standard queue must also be a standard queue.

The following attributes apply only to server-side-encryption:

- `KmsMasterKeyId` – Returns the ID of an AWS managed customer master key (CMK) for Amazon SQS or a custom CMK. For more information, see Key Terms.
- `KmsDataKeyReusePeriodSeconds` – Returns the length of time, in seconds, for which Amazon SQS can reuse a data key to encrypt or decrypt messages before calling AWS KMS again. For more information, see How Does the Data Key Reuse Period Work?.
- `SqsManagedSseEnabled` – Returns information about whether the queue is using SSE-SQS encryption using SQS owned encryption keys. Only one server-side encryption option is supported per queue (for example, SSE-KMS or SSE-SQS).

The following attributes apply only to FIFO (first-in-first-out) queues:

- `FifoQueue` – Returns information about whether the queue is FIFO. For more information, see FIFO queue logic in the *Amazon SQS Developer Guide*.

  > **Note**
  >
  > To determine whether a queue is FIFO, you can check whether `QueueName` ends with the `.fifo` suffix.

- `ContentBasedDeduplication` – Returns whether content-based deduplication is enabled for the queue. For more information, see Exactly-once processing in the *Amazon SQS Developer Guide*.

The following attributes apply only to high throughput for FIFO queues:

- `DeduplicationScope` – Specifies whether message deduplication occurs at the message group or queue level. Valid values are `messageGroup` and `queue`.
- `FifoThroughputLimit` – Specifies whether the FIFO queue throughput quota applies to the entire queue or per message group. Valid values are `perQueue` and `perMessageGroupId`. The `perMessageGroupId` value is allowed only when the value for `DeduplicationScope` is `messageGroup`.

To enable high throughput for FIFO queues, do the following:

- Set `DeduplicationScope` to `messageGroup`.
- Set `FifoThroughputLimit` to `perMessageGroupId`.

If you set these attributes to anything other than the values shown for enabling high throughput, normal throughput is in effect and deduplication occurs as specified.

For information on throughput quotas, see Quotas related to messages in the *Amazon SQS Developer Guide*.

Type: Array of strings

Valid Values: `All | Policy | VisibilityTimeout | MaximumMessageSize | MessageRetentionPeriod | ApproximateNumberOfMessages | ApproximateNumberOfMessagesNotVisible | CreatedTimestamp | LastModifiedTimestamp | QueueArn | ApproximateNumberOfMessagesDelayed | DelaySeconds | ReceiveMessageWaitTimeSeconds | RedrivePolicy | FifoQueue | ContentBasedDeduplication | KmsMasterKeyId | KmsDataKeyReusePeriodSeconds | DeduplicationScope | FifoThroughputLimit | RedriveAllowPolicy | SqsManagedSseEnabled`

Required: No

QueueUrl

The URL of the Amazon SQS queue whose attribute information is retrieved.

Queue URLs and names are case-sensitive.

Type: String

Required: Yes

Response Syntax

```
{
   "Attributes": {
      "string" : "string"
   }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Attributes

A map of attributes to their respective values.

Type: String to string map

Valid Keys: `All | Policy | VisibilityTimeout | MaximumMessageSize | MessageRetentionPeriod | ApproximateNumberOfMessages | ApproximateNumberOfMessagesNotVisible | CreatedTimestamp | LastModifiedTimestamp | QueueArn | ApproximateNumberOfMessagesDelayed | DelaySeconds | ReceiveMessageWaitTimeSeconds | RedrivePolicy | FifoQueue | ContentBasedDeduplication | KmsMasterKeyId | KmsDataKeyReusePeriodSeconds | DeduplicationScope | FifoThroughputLimit | RedriveAllowPolicy | SqsManagedSseEnabled`

Errors

For information about the errors that are common to all actions, see Common Errors.

InvalidAddress

The specified ID is invalid.

HTTP Status Code: 400

InvalidAttributeName

The specified attribute doesn't exist.

HTTP Status Code: 400

InvalidSecurity

The request was not made over HTTPS or did not use SigV4 for signing.

HTTP Status Code: 400

QueueDoesNotExist

Ensure that the `QueueUrl` is correct and that the queue has not been deleted.

HTTP Status Code: 400

RequestThrottled

The request was denied due to request throttling.

- Exceeds the permitted request rate for the queue or for the recipient of the request.
- Ensure that the request rate is within the Amazon SQS limits for sending messages. For more information, see Amazon SQS quotas in the *Amazon SQS Developer Guide*.

HTTP Status Code: 400

UnsupportedOperation

Error code 400. Unsupported operation.

HTTP Status Code: 400

Examples

The following example query request gets all the attribute values for the specified queue. The structure of AUTHPARAMS depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the *AWS General Reference*.

Example

Using AWS JSON protocol (Default)

Sample Request

```
POST / HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Target: AmazonSQS.GetQueueAtrributes
X-Amz-Date: <Date>
Content-Type: application/x-amz-json-1.0
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
{
```

```
    "QueueUrl": "https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue/",
    "AttributeNames": ["All"]
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <requestId>
Content-Length: <PayloadSizeBytes>
Date: <Date>
Content-Type: application/x-amz-json-1.0
{
    "Attributes": {
       "QueueArn": "arn:aws:sqs:us-east-1:555555555555:MyQueue",
        "ApproximateNumberOfMessages": "0",
        "ApproximateNumberOfMessagesNotVisible": "0",
        "ApproximateNumberOfMessagesDelayed": "0",
        "CreatedTimestamp": "1676665337",
        "LastModifiedTimestamp": "1677096375",
        "VisibilityTimeout": "60",
        "MaximumMessageSize": "12345",
        "MessageRetentionPeriod": "345600",
        "DelaySeconds": "0",
        "Policy": "{\"Version\":\"2012-10-17\",\"Id\":\"Policy1677095510157\",
\"Statement\":[{\"Sid\":\"Stmt1677095506939\",\"Effect\":\"Allow\",\"Principal
\":\"*\",\"Action\":\"sqs:ReceiveMessage\",\"Resource\":\"arn:aws:sqs:us-
east-1:555555555555:MyQueue6\"}]}",
        "RedriveAllowPolicy": "{\"redrivePermission\":\"allowAll\"}",
        "ReceiveMessageWaitTimeSeconds": "2",
        "SqsManagedSseEnabled": "true"
    }
}
```

Example

Using AWS query protocol

Sample Request

```
POST /177715257436/MyQueue/ HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
Content-Type: application/x-www-form-urlencoded
X-Amz-Date: <Date>
```

```
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
Action=GetQueueAttributes
&AttributeName.1=All
```

Sample Response

```
HTTP/1.1 200 OK
<?xml version="1.0"?>
<GetQueueAttributesResponse xmlns="http://queue.amazonaws.com/doc/2012-11-05/">
    <GetQueueAttributesResult>
        <Attribute>
            <Name>QueueArn</Name>
            <Value>arn:aws:sqs:us-east-1:555555555555:MyQueue</Value>
        </Attribute>
        <Attribute>
            <Name>ApproximateNumberOfMessages</Name>
            <Value>5</Value>
        </Attribute>
        <Attribute>
            <Name>ApproximateNumberOfMessagesNotVisible</Name>
            <Value>0</Value>
        </Attribute>
        <Attribute>
            <Name>ApproximateNumberOfMessagesDelayed</Name>
            <Value>0</Value>
        </Attribute>
        <Attribute>
            <Name>CreatedTimestamp</Name>
            <Value>1677110910</Value>
        </Attribute>
        <Attribute>
            <Name>LastModifiedTimestamp</Name>
            <Value>1677110910</Value>
        </Attribute>
        <Attribute>
            <Name>VisibilityTimeout</Name>
            <Value>40</Value>
        </Attribute>
        <Attribute>
            <Name>MaximumMessageSize</Name>
            <Value>262144</Value>
```

```
        </Attribute>
        <Attribute>
            <Name>MessageRetentionPeriod</Name>
            <Value>345600</Value>
        </Attribute>
        <Attribute>
            <Name>DelaySeconds</Name>
            <Value>0</Value>
        </Attribute>
        <Attribute>
            <Name>ReceiveMessageWaitTimeSeconds</Name>
            <Value>0</Value>
        </Attribute>
        <Attribute>
            <Name>SqsManagedSseEnabled</Name>
            <Value>true</Value>
        </Attribute>
    </GetQueueAttributesResult>
    <ResponseMetadata>
        <RequestId>1cffc414-8cb4-54a8-9519-98644ca5f987</RequestId>
    </ResponseMetadata>
</GetQueueAttributesResponse>
```

Example

The following example query request gets three attribute values for the specified queue. The structure of AUTHPARAMS depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the *AWS General Reference*.

Using AWS JSON protocol (Default)

Sample Request

```
POST / HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Target: AmazonSQS.GetQueueAtrributes
X-Amz-Date: <Date>
Content-Type: application/x-amz-json-1.0
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
{
    "QueueUrl": "https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue/",
```

```
    "AttributeNames": ["VisibilityTimeout", "DelaySeconds",
 "ReceiveMessageWaitTimeSeconds"]
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <requestId>
Content-Length: <PayloadSizeBytes>
Date: <Date>
Content-Type: application/x-amz-json-1.0
{
    "Attributes": {
       "VisibilityTimeout": "35",
       "DelaySeconds": "45",
        "ReceiveMessageWaitTimeSeconds": "20"
    }
}
```

Example

Using AWS query protocol

Sample Request

```
POST /177715257436/MyQueue/ HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
Content-Type: application/x-www-form-urlencoded
X-Amz-Date: <Date>
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
Action=GetQueueAttributes
&AttributeName.1=VisibilityTimeout
&AttributeName.2=DelaySeconds
&AttributeName.3=ReceiveMessageWaitTimeSeconds
```

Sample Response

```
HTTP/1.1 200 OK
<?xml version="1.0"?>
<GetQueueAttributesResponse xmlns="http://queue.amazonaws.com/doc/2012-11-05/">
```

```
    <GetQueueAttributesResult>
        <Attribute>
            <Name>VisibilityTimeout</Name>
            <Value>35</Value>
        </Attribute>
        <Attribute>
            <Name>DelaySeconds</Name>
            <Value>45</Value>
        </Attribute>
        <Attribute>
            <Name>ReceiveMessageWaitTimeSeconds</Name>
            <Value>20</Value>
        </Attribute>
    </GetQueueAttributesResult>
    <ResponseMetadata>
        <RequestId>60462930-c7fd-5ef8-b6a0-75a20b5e17b8</RequestId>
    </ResponseMetadata>
</GetQueueAttributesResponse>
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python (Boto3)
- AWS SDK for Ruby V3

GetQueueUrl

The `GetQueueUrl` API returns the URL of an existing Amazon SQS queue. This is useful when you know the queue's name but need to retrieve its URL for further operations.

To access a queue owned by another AWS account, use the `QueueOwnerAWSAccountId` parameter to specify the account ID of the queue's owner. Note that the queue owner must grant you the necessary permissions to access the queue. For more information about accessing shared queues, see the `AddPermission` API or Allow developers to write messages to a shared queue in the *Amazon SQS Developer Guide*.

Request Syntax

```
{
   "QueueName": "string",
   "QueueOwnerAWSAccountId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

QueueName

(Required) The name of the queue for which you want to fetch the URL. The name can be up to 80 characters long and can include alphanumeric characters, hyphens (-), and underscores (_). Queue URLs and names are case-sensitive.

Type: String

Required: Yes

QueueOwnerAWSAccountId

(Optional) The AWS account ID of the account that created the queue. This is only required when you are attempting to access a queue owned by another AWS account.

Type: String

Required: No

Response Syntax

```
{
   "QueueUrl": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

QueueUrl

The URL of the queue.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

InvalidAddress

The specified ID is invalid.

HTTP Status Code: 400

InvalidSecurity

The request was not made over HTTPS or did not use SigV4 for signing.

HTTP Status Code: 400

QueueDoesNotExist

Ensure that the `QueueUrl` is correct and that the queue has not been deleted.

HTTP Status Code: 400

RequestThrottled

The request was denied due to request throttling.

- Exceeds the permitted request rate for the queue or for the recipient of the request.
- Ensure that the request rate is within the Amazon SQS limits for sending messages. For more information, see Amazon SQS quotas in the *Amazon SQS Developer Guide*.

HTTP Status Code: 400

UnsupportedOperation

Error code 400. Unsupported operation.

HTTP Status Code: 400

Examples

The following are examples of using the `GetQueueUrl` API via both JSON and query protocols. The structure of `AUTHPARAMS` depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the *AWS General Reference*.

Example

Using AWS JSON protocol (Default)

Sample Request

```
POST / HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Target: AmazonSQS.GetQueueUrl
X-Amz-Date: <Date>
Content-Type: application/x-amz-json-1.0
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
{
    "QueueName": "MyQueue"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <requestId>
Content-Length: <PayloadSizeBytes>
```

```
Date: <Date>
Content-Type: application/x-amz-json-1.0
{
    "QueueUrl": "https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue"
}
```

Example

Using AWS query protocol

Sample Request

```
POST / HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
Content-Type: application/x-www-form-urlencoded
X-Amz-Date: <Date>
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
Action=GetQueueUrl
&QueueName=MyQueue
```

Sample Response

```
HTTP/1.1 200 OK
<?xml version="1.0"?>
<GetQueueUrlResponse xmlns="http://queue.amazonaws.com/doc/2012-11-05/">
    <GetQueueUrlResult>
        <QueueUrl>https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue</QueueUrl>
    </GetQueueUrlResult>
    <ResponseMetadata>
        <RequestId>552d6f30-4c8e-5b32-aaed-33408c7d6c38</RequestId>
    </ResponseMetadata>
</GetQueueUrlResponse>
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface V2

- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python (Boto3)
- AWS SDK for Ruby V3

ListDeadLetterSourceQueues

Returns a list of your queues that have the `RedrivePolicy` queue attribute configured with a dead-letter queue.

The `ListDeadLetterSourceQueues` methods supports pagination. Set parameter `MaxResults` in the request to specify the maximum number of results to be returned in the response. If you do not set `MaxResults`, the response includes a maximum of 1,000 results. If you set `MaxResults` and there are additional results to display, the response includes a value for `NextToken`. Use `NextToken` as a parameter in your next request to `ListDeadLetterSourceQueues` to receive the next page of results.

For more information about using dead-letter queues, see Using Amazon SQS Dead-Letter Queues in the *Amazon SQS Developer Guide*.

Request Syntax

```
{
   "MaxResults": number,
   "NextToken": "string",
   "QueueUrl": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

MaxResults

Maximum number of results to include in the response. Value range is 1 to 1000. You must set `MaxResults` to receive a value for `NextToken` in the response.

Type: Integer

Required: No

NextToken

Pagination token to request the next set of results.

Type: String

Required: No

QueueUrl

The URL of a dead-letter queue.

Queue URLs and names are case-sensitive.

Type: String

Required: Yes

Response Syntax

```
{
   "NextToken": "string",
   "queueUrls": [ "string" ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

NextToken

Pagination token to include in the next request. Token value is `null` if there are no additional results to request, or if you did not set `MaxResults` in the request.

Type: String

queueUrls

A list of source queue URLs that have the `RedrivePolicy` queue attribute configured with a dead-letter queue.

Type: Array of strings

Errors

For information about the errors that are common to all actions, see Common Errors.

InvalidAddress

The specified ID is invalid.

HTTP Status Code: 400

InvalidSecurity

The request was not made over HTTPS or did not use SigV4 for signing.

HTTP Status Code: 400

QueueDoesNotExist

Ensure that the `QueueUrl` is correct and that the queue has not been deleted.

HTTP Status Code: 400

RequestThrottled

The request was denied due to request throttling.

- Exceeds the permitted request rate for the queue or for the recipient of the request.
- Ensure that the request rate is within the Amazon SQS limits for sending messages. For more information, see Amazon SQS quotas in the *Amazon SQS Developer Guide*.

HTTP Status Code: 400

UnsupportedOperation

Error code 400. Unsupported operation.

HTTP Status Code: 400

Examples

The following example query request returns a list of dead letter source queues. In this example, only one source queue, `MySourceQueue`, is configured with a dead-letter queue. The structure of `AUTHPARAMS` depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the *AWS General Reference*.

Example

Using AWS JSON protocol (Default)

Sample Request

```
POST / HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Target: AmazonSQS.ListDeadLetterSourceQueues
X-Amz-Date: <Date>
Content-Type: application/x-amz-json-1.0
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
{
    "QueueUrl": "https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <requestId>
Content-Length: <PayloadSizeBytes>
Date: <Date>
Content-Type: application/x-amz-json-1.0
{
    "queueUrls": [
        "https://sqs.us-east-1.amazonaws.com/123456789012/MySourceQueue</QueueUrl>"
    ]
}
```

Example

Using AWS query protocol

Sample Request

```
POST /177715257436/MyQueue/ HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
Content-Type: application/x-www-form-urlencoded
X-Amz-Date: <Date>
Authorization: <AuthParams>
```

```
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
Action=ListDeadLetterSourceQueues
```

Sample Response

```
HTTP/1.1 200 OK
<?xml version="1.0"?>
<ListDeadLetterSourceQueuesResponse xmlns="http://queue.amazonaws.com/doc/2012-11-05/">
    <ListDeadLetterSourceQueuesResult/>
    <ResponseMetadata>
        <RequestId>8e70ce7e-e3d4-5109-9204-920e21745daf</RequestId>
    </ResponseMetadata>
</ListDeadLetterSourceQueuesResponse>
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python (Boto3)
- AWS SDK for Ruby V3

ListMessageMoveTasks

Gets the most recent message movement tasks (up to 10) under a specific source queue.

Note

- This action is currently limited to supporting message redrive from dead-letter queues (DLQs) only. In this context, the source queue is the dead-letter queue (DLQ), while the destination queue can be the original source queue (from which the messages were driven to the dead-letter-queue), or a custom destination queue.
- Only one active message movement task is supported per queue at any given time.

Request Syntax

```
{
   "MaxResults": number,
   "SourceArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

MaxResults

The maximum number of results to include in the response. The default is 1, which provides the most recent message movement task. The upper limit is 10.

Type: Integer

Required: No

SourceArn

The ARN of the queue whose message movement tasks are to be listed.

Type: String

Required: Yes

Response Syntax

```
{
   "Results": [
      {
         "ApproximateNumberOfMessagesMoved": number,
         "ApproximateNumberOfMessagesToMove": number,
         "DestinationArn": "string",
         "FailureReason": "string",
         "MaxNumberOfMessagesPerSecond": number,
         "SourceArn": "string",
         "StartedTimestamp": number,
         "Status": "string",
         "TaskHandle": "string"
      }
   ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Results

A list of message movement tasks and their attributes.

Type: Array of ListMessageMoveTasksResultEntry objects

Errors

For information about the errors that are common to all actions, see Common Errors.

InvalidAddress

The specified ID is invalid.

HTTP Status Code: 400

InvalidSecurity

The request was not made over HTTPS or did not use SigV4 for signing.

HTTP Status Code: 400

RequestThrottled

The request was denied due to request throttling.

- Exceeds the permitted request rate for the queue or for the recipient of the request.
- Ensure that the request rate is within the Amazon SQS limits for sending messages. For more information, see Amazon SQS quotas in the *Amazon SQS Developer Guide*.

HTTP Status Code: 400

ResourceNotFoundException

One or more specified resources don't exist.

HTTP Status Code: 400

UnsupportedOperation

Error code 400. Unsupported operation.

HTTP Status Code: 400

Examples

Example

Using AWS query protocol

The following example query lists the five most recent message movement tasks on `MyDeadLetterQueue`, which has only two tasks since it was created. The structure of `AUTHPARAMS` depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the *AWS General Reference*.

Sample Request

```
POST / HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
```

```
X-Amz-Date: <Date>
Content-Type: application/x-www-form-urlencoded
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
Action=ListMessageMoveTasks
&SourceArn=arn:aws:sqs:us-east-1:555555555555:MyDeadLetterQueue
&MaxResults=5
```

Sample Response

```
HTTP/1.1 200 OK
<?xml version="1.0"?>
<ListMessageMoveTasksResponse xmlns="http://queue.amazonaws.com/doc/2012-11-05/">
    <ListMessageMoveTasksResult>
        <Result>
            <ApproximateNumberOfMessagesMoved>50</ApproximateNumberOfMessagesMoved>
            <ApproximateNumberOfMessagesToMove>0</ApproximateNumberOfMessagesToMove>
            <DestinationArn>arn:aws:sqs:us-east-1:555555555555:MySourceQueue</
DestinationArn>
            <MaxNumberOfMessagesPerSecond>20</MaxNumberOfMessagesPerSecond>
            <SourceArn>arn:aws:sqs:us-east-1:555555555555:MyDeadLetterQueue</SourceArn>
            <StartedTimestamp>1684429053010</StartedTimestamp>
            <Status>COMPLETED</Status>
        </Result>
        <Result>
            <ApproximateNumberOfMessagesMoved>50</ApproximateNumberOfMessagesMoved>
            <ApproximateNumberOfMessagesToMove>100</ApproximateNumberOfMessagesToMove>
            <DestinationArn>arn:aws:sqs:us-east-1:555555555555:MySourceQueue</
DestinationArn>
            <MaxNumberOfMessagesPerSecond>10</MaxNumberOfMessagesPerSecond>
            <SourceArn>arn:aws:sqs:us-east-1:555555555555:MyDeadLetterQueue</SourceArn>
            <StartedTimestamp>1684429993999</StartedTimestamp>
            <Status>RUNNING</Status>

 <TaskHandle>eyJ0YXNrSWQiOiJkYzE2OWUwNC0wZTU1LTQ0ZDItYWE5MC1jMDgwY2ExZjM2ZjciLCJzb3VyY2VBcm4iOi
TaskHandle>
        </Result>
        <ResponseMetadata>
            <RequestId>9b20926c-8b35-5d8e-9559-ce1c22e754dc</RequestId>
        </ResponseMetadata>
    </ListMessageMoveTasksResult>
</ListMessageMoveTasksResponse>
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python (Boto3)
- AWS SDK for Ruby V3

ListQueues

Returns a list of your queues in the current region. The response includes a maximum of 1,000 results. If you specify a value for the optional `QueueNamePrefix` parameter, only queues with a name that begins with the specified value are returned.

The `listQueues` methods supports pagination. Set parameter `MaxResults` in the request to specify the maximum number of results to be returned in the response. If you do not set `MaxResults`, the response includes a maximum of 1,000 results. If you set `MaxResults` and there are additional results to display, the response includes a value for `NextToken`. Use `NextToken` as a parameter in your next request to `listQueues` to receive the next page of results.

> **Note**
>
> Cross-account permissions don't apply to this action. For more information, see Grant cross-account permissions to a role and a username in the *Amazon SQS Developer Guide*.

Request Syntax

```
{
   "MaxResults": number,
   "NextToken": "string",
   "QueueNamePrefix": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

MaxResults

Maximum number of results to include in the response. Value range is 1 to 1000. You must set `MaxResults` to receive a value for `NextToken` in the response.

Type: Integer

Required: No

NextToken

Pagination token to request the next set of results.

Type: String

Required: No

QueueNamePrefix

A string to use for filtering the list results. Only those queues whose name begins with the specified string are returned.

Queue URLs and names are case-sensitive.

Type: String

Required: No

Response Syntax

```
{
   "NextToken": "string",
   "QueueUrls": [ "string" ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

NextToken

Pagination token to include in the next request. Token value is `null` if there are no additional results to request, or if you did not set `MaxResults` in the request.

Type: String

QueueUrls

A list of queue URLs, up to 1,000 entries, or the value of `MaxResults` that you sent in the request.

Type: Array of strings

Errors

For information about the errors that are common to all actions, see Common Errors.

InvalidAddress

The specified ID is invalid.

HTTP Status Code: 400

InvalidSecurity

The request was not made over HTTPS or did not use SigV4 for signing.

HTTP Status Code: 400

RequestThrottled

The request was denied due to request throttling.

- Exceeds the permitted request rate for the queue or for the recipient of the request.
- Ensure that the request rate is within the Amazon SQS limits for sending messages. For more information, see Amazon SQS quotas in the *Amazon SQS Developer Guide*.

HTTP Status Code: 400

UnsupportedOperation

Error code 400. Unsupported operation.

HTTP Status Code: 400

Examples

The following example query request returns the queues whose names begin with the letter `t` The structure of `AUTHPARAMS` depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the *AWS General Reference*.

Example

Using AWS JSON protocol (Default)

Sample Request

```
POST / HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Target: AmazonSQS.ListQueues
X-Amz-Date: <Date>
Content-Type: application/x-amz-json-1.0
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
{
    "QueueNamePrefix": "My"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <requestId>
Content-Length: <PayloadSizeBytes>
Date: <Date>
Content-Type: application/x-amz-json-1.0
{
    "QueueUrls": [
        "https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue",
        "https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue1648169377027",
        "https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue1648169549830",
        "https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue1648227401019",
        "https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue1648248132466",
        "https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue1649201932174",
        "https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue2"
    ]
}
```

Example

Using AWS query protocol

Sample Request

```
POST / HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
Content-Type: application/x-www-form-urlencoded
```

```
X-Amz-Date: <Date>
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
Action=ListQueues&QueueNamePrefix=M
```

Sample Response

```
HTTP/1.1 200 OK
<?xml version="1.0"?>
<ListQueuesResponse xmlns="http://queue.amazonaws.com/doc/2012-11-05/">
    <ListQueuesResult>
        <QueueUrl>https://sqs.us-east-1.amazonaws.com/177715257436/
MyQueue1648169377027</QueueUrl>
        <QueueUrl>https://sqs.us-east-1.amazonaws.com/177715257436/
MyQueue1648169549830</QueueUrl>
        <QueueUrl>https://sqs.us-east-1.amazonaws.com/177715257436/
MyQueue1648227401019</QueueUrl>
        <QueueUrl>https://sqs.us-east-1.amazonaws.com/177715257436/
MyQueue1648248132466</QueueUrl>
        <QueueUrl>https://sqs.us-east-1.amazonaws.com/177715257436/
MyQueue1649201932174</QueueUrl>
        <QueueUrl>https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue22</QueueUrl>
        <QueueUrl>https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue23</QueueUrl>
        <QueueUrl>https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue233</
QueueUrl>
        <QueueUrl>https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue5</QueueUrl>
        <QueueUrl>https://sqs.us-east-1.amazonaws.com/177715257436/MyQueueTest</
QueueUrl>
    </ListQueuesResult>
    <ResponseMetadata>
        <RequestId>f525e5e2-86cd-5d1b-aee9-b992443254c0</RequestId>
    </ResponseMetadata>
</ListQueuesResponse>
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface V2
- AWS SDK for .NET V4

- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python (Boto3)
- AWS SDK for Ruby V3

ListQueueTags

List all cost allocation tags added to the specified Amazon SQS queue. For an overview, see Tagging Your Amazon SQS Queues in the *Amazon SQS Developer Guide*.

> **Note**
>
> Cross-account permissions don't apply to this action. For more information, see Grant cross-account permissions to a role and a username in the *Amazon SQS Developer Guide*.

Request Syntax

```
{
   "QueueUrl": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

QueueUrl

The URL of the queue.

Type: String

Required: Yes

Response Syntax

```
{
   "Tags": {
      "string" : "string"
   }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Tags

The list of all tags added to the specified queue.

Type: String to string map

Errors

For information about the errors that are common to all actions, see Common Errors.

InvalidAddress

The specified ID is invalid.

HTTP Status Code: 400

InvalidSecurity

The request was not made over HTTPS or did not use SigV4 for signing.

HTTP Status Code: 400

QueueDoesNotExist

Ensure that the `QueueUrl` is correct and that the queue has not been deleted.

HTTP Status Code: 400

RequestThrottled

The request was denied due to request throttling.

- Exceeds the permitted request rate for the queue or for the recipient of the request.
- Ensure that the request rate is within the Amazon SQS limits for sending messages. For more information, see Amazon SQS quotas in the *Amazon SQS Developer Guide*.

HTTP Status Code: 400

UnsupportedOperation

Error code 400. Unsupported operation.

HTTP Status Code: 400

Examples

This example illustrates one usage of `ListQueueTags`.

Example

Using AWS JSON protocol (Default)

Sample Request

```
POST / HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Target: AmazonSQS.ListQueueTags
X-Amz-Date: <Date>
Content-Type: application/x-amz-json-1.0
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
{
    "QueueUrl": "https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue/"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <requestId>
Content-Length: <PayloadSizeBytes>
Date: <Date>
Content-Type: application/x-amz-json-1.0
{
    "Tags": {
        "QueueType": "Production"
    }
}
```

Example

Using AWS query protocol

Sample Request

```
POST /177715257436/MyQueue HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
Content-Type: application/x-www-form-urlencoded
X-Amz-Date: <Date>
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
Action=ListQueueTags
```

Sample Response

```
HTTP/1.1 200 OK
<?xml version="1.0"?>
<ListQueueTagsResponse xmlns="http://queue.amazonaws.com/doc/2012-11-05/">
    <ListQueueTagsResult>
        <Tag>
            <Key>QueueType</Key>
            <Value>Production</Value>
        </Tag>
    </ListQueueTagsResult>
    <ResponseMetadata>
        <RequestId>02c89a6b-9fc0-564a-9ed1-c61b5cacdc6d</RequestId>
    </ResponseMetadata>
</ListQueueTagsResponse>
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2

- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python (Boto3)
- AWS SDK for Ruby V3

PurgeQueue

Deletes available messages in a queue (including in-flight messages) specified by the QueueURL parameter.

> **Important**
>
> When you use the PurgeQueue action, you can't retrieve any messages deleted from a queue.
> The message deletion process takes up to 60 seconds. We recommend waiting for 60 seconds regardless of your queue's size.

Messages sent to the queue *before* you call PurgeQueue might be received but are deleted within the next minute.

Messages sent to the queue *after* you call PurgeQueue might be deleted while the queue is being purged.

Request Syntax

```
{
   "QueueUrl": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

QueueUrl

The URL of the queue from which the PurgeQueue action deletes messages.

Queue URLs and names are case-sensitive.

Type: String

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors.

InvalidAddress

The specified ID is invalid.

HTTP Status Code: 400

InvalidSecurity

The request was not made over HTTPS or did not use SigV4 for signing.

HTTP Status Code: 400

PurgeQueueInProgress

Indicates that the specified queue previously received a `PurgeQueue` request within the last 60 seconds (the time it can take to delete the messages in the queue).

HTTP Status Code: 400

QueueDoesNotExist

Ensure that the `QueueUrl` is correct and that the queue has not been deleted.

HTTP Status Code: 400

RequestThrottled

The request was denied due to request throttling.

- Exceeds the permitted request rate for the queue or for the recipient of the request.
- Ensure that the request rate is within the Amazon SQS limits for sending messages. For more information, see Amazon SQS quotas in the *Amazon SQS Developer Guide*.

HTTP Status Code: 400

UnsupportedOperation

Error code 400. Unsupported operation.

HTTP Status Code: 400

Examples

The following example query request purges a queue named MyQueue. The structure of AUTHPARAMS depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the *AWS General Reference*.

Example

Using AWS JSON protocol (Default)

Sample Request

```
POST / HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Target: AmazonSQS.PurgeQueue
X-Amz-Date: <Date>
Content-Type: application/x-amz-json-1.0
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
{
    "QueueUrl": "https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue/"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <requestId>
Content-Length: 0
Date: <Date>
Content-Type: application/x-amz-json-1.0
```

Example

Using AWS query protocol

Sample Request

```
POST /177715257436/MyQueue HTTP/1.1
```

```
Host: sqs.us-east-1.amazonaws.com
Content-Type: application/x-www-form-urlencoded
X-Amz-Date: <Date>
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive

Action=PurgeQueue
```

Sample Response

```
HTTP/1.1 200 OK
<?xml version="1.0"?>
<PurgeQueueResponse xmlns="http://queue.amazonaws.com/doc/2012-11-05/">
    <ResponseMetadata>
        <RequestId>706cb8e1-8799-5ef8-9472-e4914393f2f0</RequestId>
    </ResponseMetadata>
</PurgeQueueResponse>
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python (Boto3)
- AWS SDK for Ruby V3

ReceiveMessage

Retrieves one or more messages (up to 10), from the specified queue. Using the `WaitTimeSeconds` parameter enables long-poll support. For more information, see Amazon SQS Long Polling in the *Amazon SQS Developer Guide*.

Short poll is the default behavior where a weighted random set of machines is sampled on a `ReceiveMessage` call. Therefore, only the messages on the sampled machines are returned. If the number of messages in the queue is small (fewer than 1,000), you most likely get fewer messages than you requested per `ReceiveMessage` call. If the number of messages in the queue is extremely small, you might not receive any messages in a particular `ReceiveMessage` response. If this happens, repeat the request.

For each message returned, the response includes the following:

- The message body.
- An MD5 digest of the message body. For information about MD5, see RFC1321.
- The `MessageId` you received when you sent the message to the queue.
- The receipt handle.
- The message attributes.
- An MD5 digest of the message attributes.

The receipt handle is the identifier you must provide when deleting the message. For more information, see Queue and Message Identifiers in the *Amazon SQS Developer Guide*.

You can provide the `VisibilityTimeout` parameter in your request. The parameter is applied to the messages that Amazon SQS returns in the response. If you don't include the parameter, the overall visibility timeout for the queue is used for the returned messages. The default visibility timeout for a queue is 30 seconds.

> **Note**
>
> In the future, new attributes might be added. If you write code that calls this action, we recommend that you structure your code so that it can handle new attributes gracefully.

Request Syntax

```
{
   "AttributeNames": [ "string" ],
   "MaxNumberOfMessages": number,
   "MessageAttributeNames": [ "string" ],
   "MessageSystemAttributeNames": [ "string" ],
   "QueueUrl": "string",
   "ReceiveRequestAttemptId": "string",
   "VisibilityTimeout": number,
   "WaitTimeSeconds": number
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

AttributeNames

This parameter has been deprecated.

> **Important**
>
> This parameter has been discontinued but will be supported for backward compatibility. To provide attribute names, you are encouraged to use `MessageSystemAttributeNames`.

A list of attributes that need to be returned along with each message. These attributes include:

- `All` – Returns all values.
- `ApproximateFirstReceiveTimestamp` – Returns the time the message was first received from the queue (epoch time in milliseconds).
- `ApproximateReceiveCount` – Returns the number of times a message has been received across all queues but not deleted.
- `AWSTraceHeader` – Returns the AWS X-Ray trace header string.
- `SenderId`
  - For a user, returns the user ID, for example `ABCDEFGHI1JKLMNOPQ23R`.

  - For an IAM role, returns the IAM role ID, for example `ABCDE1F2GH3I4JK5LMNOP:i-a123b456`.
- `SentTimestamp` – Returns the time the message was sent to the queue (epoch time in milliseconds).
- `SqsManagedSseEnabled` – Enables server-side queue encryption using SQS owned encryption keys. Only one server-side encryption option is supported per queue (for example, SSE-KMS or SSE-SQS).
- `MessageDeduplicationId` – Returns the value provided by the producer that calls the `SendMessage` action.
- `MessageGroupId` – Returns the value provided by the producer that calls the `SendMessage` action.
- `SequenceNumber` – Returns the value provided by Amazon SQS.

Type: Array of strings

Valid Values: `All | Policy | VisibilityTimeout | MaximumMessageSize | MessageRetentionPeriod | ApproximateNumberOfMessages | ApproximateNumberOfMessagesNotVisible | CreatedTimestamp | LastModifiedTimestamp | QueueArn | ApproximateNumberOfMessagesDelayed | DelaySeconds | ReceiveMessageWaitTimeSeconds | RedrivePolicy | FifoQueue | ContentBasedDeduplication | KmsMasterKeyId | KmsDataKeyReusePeriodSeconds | DeduplicationScope | FifoThroughputLimit | RedriveAllowPolicy | SqsManagedSseEnabled`

Required: No

MaxNumberOfMessages

The maximum number of messages to return. Amazon SQS never returns more messages than this value (however, fewer messages might be returned). Valid values: 1 to 10. Default: 1.

Type: Integer

Required: No

MessageAttributeNames

The name of the message attribute, where *N* is the index.

- The name can contain alphanumeric characters and the underscore (_), hyphen (-), and period (.).

- The name is case-sensitive and must be unique among all attribute names for the message.
- The name must not start with AWS-reserved prefixes such as `AWS.` or `Amazon.` (or any casing variants).
- The name must not start or end with a period (`.`), and it should not have periods in succession (`..`).
- The name can be up to 256 characters long.

When using `ReceiveMessage`, you can send a list of attribute names to receive, or you can return all of the attributes by specifying `All` or `.*` in your request. You can also use all message attributes starting with a prefix, for example `bar.*`.

Type: Array of strings

Required: No

MessageSystemAttributeNames

A list of attributes that need to be returned along with each message. These attributes include:

- `All` – Returns all values.
- `ApproximateFirstReceiveTimestamp` – Returns the time the message was first received from the queue (epoch time in milliseconds).
- `ApproximateReceiveCount` – Returns the number of times a message has been received across all queues but not deleted.
- `AWSTraceHeader` – Returns the AWS X-Ray trace header string.
- `SenderId`
  - For a user, returns the user ID, for example `ABCDEFGHI1JKLMNOPQ23R`.
  - For an IAM role, returns the IAM role ID, for example `ABCDE1F2GH3I4JK5LMNOP:i-a123b456`.
- `SentTimestamp` – Returns the time the message was sent to the queue (epoch time in milliseconds).
- `SqsManagedSseEnabled` – Enables server-side queue encryption using SQS owned encryption keys. Only one server-side encryption option is supported per queue (for example, SSE-KMS or SSE-SQS).
- `MessageDeduplicationId` – Returns the value provided by the producer that calls the `SendMessage` action.

- `MessageGroupId` – Returns the value provided by the producer that calls the `SendMessage` action.
- `SequenceNumber` – Returns the value provided by Amazon SQS.

Type: Array of strings

Valid Values: `All | SenderId | SentTimestamp | ApproximateReceiveCount | ApproximateFirstReceiveTimestamp | SequenceNumber | MessageDeduplicationId | MessageGroupId | AWSTraceHeader | DeadLetterQueueSourceArn`

Required: No

QueueUrl

The URL of the Amazon SQS queue from which messages are received.

Queue URLs and names are case-sensitive.

Type: String

Required: Yes

ReceiveRequestAttemptId

This parameter applies only to FIFO (first-in-first-out) queues.

The token used for deduplication of `ReceiveMessage` calls. If a networking issue occurs after a `ReceiveMessage` action, and instead of a response you receive a generic error, it is possible to retry the same action with an identical `ReceiveRequestAttemptId` to retrieve the same set of messages, even if their visibility timeout has not yet expired.

- You can use `ReceiveRequestAttemptId` only for 5 minutes after a `ReceiveMessage` action.
- When you set `FifoQueue`, a caller of the `ReceiveMessage` action can provide a `ReceiveRequestAttemptId` explicitly.
- It is possible to retry the `ReceiveMessage` action with the same `ReceiveRequestAttemptId` if none of the messages have been modified (deleted or had their visibility changes).
- During a visibility timeout, subsequent calls with the same `ReceiveRequestAttemptId` return the same messages and receipt handles. If a retry occurs within the deduplication

interval, it resets the visibility timeout. For more information, see Visibility Timeout in the *Amazon SQS Developer Guide*.

> **Important**
>
> If a caller of the `ReceiveMessage` action still processes messages when the visibility timeout expires and messages become visible, another worker consuming from the same queue can receive the same messages and therefore process duplicates. Also, if a consumer whose message processing time is longer than the visibility timeout tries to delete the processed messages, the action fails with an error.
>
> To mitigate this effect, ensure that your application observes a safe threshold before the visibility timeout expires and extend the visibility timeout as necessary.

- While messages with a particular `MessageGroupId` are invisible, no more messages belonging to the same `MessageGroupId` are returned until the visibility timeout expires. You can still receive messages with another `MessageGroupId` from your FIFO queue as long as they are visible.
- If a caller of `ReceiveMessage` can't track the `ReceiveRequestAttemptId`, no retries work until the original visibility timeout expires. As a result, delays might occur but the messages in the queue remain in a strict order.

The maximum length of `ReceiveRequestAttemptId` is 128 characters. `ReceiveRequestAttemptId` can contain alphanumeric characters (`a-z`, `A-Z`, `0-9`) and punctuation (`!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~`).

For best practices of using `ReceiveRequestAttemptId`, see Using the ReceiveRequestAttemptId Request Parameter in the *Amazon SQS Developer Guide*.

Type: String

Required: No

VisibilityTimeout

The duration (in seconds) that the received messages are hidden from subsequent retrieve requests after being retrieved by a `ReceiveMessage` request. If not specified, the default visibility timeout for the queue is used, which is 30 seconds.

Understanding `VisibilityTimeout`:

- When a message is received from a queue, it becomes temporarily invisible to other consumers for the duration of the visibility timeout. This prevents multiple consumers from processing the same message simultaneously. If the message is not deleted or its visibility timeout is not extended before the timeout expires, it becomes visible again and can be retrieved by other consumers.
- Setting an appropriate visibility timeout is crucial. If it's too short, the message might become visible again before processing is complete, leading to duplicate processing. If it's too long, it delays the reprocessing of messages if the initial processing fails.
- You can adjust the visibility timeout using the `--visibility-timeout` parameter in the `receive-message` command to match the processing time required by your application.
- A message that isn't deleted or a message whose visibility isn't extended before the visibility timeout expires counts as a failed receive. Depending on the configuration of the queue, the message might be sent to the dead-letter queue.

For more information, see Visibility Timeout in the *Amazon SQS Developer Guide*.

Type: Integer

Required: No

WaitTimeSeconds

The duration (in seconds) for which the call waits for a message to arrive in the queue before returning. If a message is available, the call returns sooner than `WaitTimeSeconds`. If no messages are available and the wait time expires, the call does not return a message list. If you are using the Java SDK, it returns a `ReceiveMessageResponse` object, which has a empty list instead of a Null object.

> **Important**
>
> To avoid HTTP errors, ensure that the HTTP response timeout for `ReceiveMessage` requests is longer than the `WaitTimeSeconds` parameter. For example, with the Java SDK, you can set HTTP transport settings using the NettyNioAsyncHttpClient for asynchronous clients, or the ApacheHttpClient for synchronous clients.

Type: Integer

Required: No

Response Syntax

```
{
   "Messages": [
      {
         "Attributes": {
            "string" : "string"
         },
         "Body": "string",
         "MD5OfBody": "string",
         "MD5OfMessageAttributes": "string",
         "MessageAttributes": {
            "string" : {
               "BinaryListValues": [ blob ],
               "BinaryValue": blob,
               "DataType": "string",
               "StringListValues": [ "string" ],
               "StringValue": "string"
            }
         },
         "MessageId": "string",
         "ReceiptHandle": "string"
      }
   ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Messages

A list of messages.

Type: Array of Message objects

Errors

For information about the errors that are common to all actions, see Common Errors.

InvalidAddress

The specified ID is invalid.

HTTP Status Code: 400

InvalidSecurity

The request was not made over HTTPS or did not use SigV4 for signing.

HTTP Status Code: 400

KmsAccessDenied

The caller doesn't have the required KMS access.

HTTP Status Code: 400

KmsDisabled

The request was denied due to request throttling.

HTTP Status Code: 400

KmsInvalidKeyUsage

The request was rejected for one of the following reasons:

- The KeyUsage value of the KMS key is incompatible with the API operation.
- The encryption algorithm or signing algorithm specified for the operation is incompatible with the type of key material in the KMS key (KeySpec).

HTTP Status Code: 400

KmsInvalidState

The request was rejected because the state of the specified resource is not valid for this request.

HTTP Status Code: 400

KmsNotFound

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 400

KmsOptInRequired

The request was rejected because the specified key policy isn't syntactically or semantically correct.

HTTP Status Code: 400

KmsThrottled

AWS KMS throttles requests for the following conditions.

HTTP Status Code: 400

OverLimit

The specified action violates a limit. For example, `ReceiveMessage` returns this error if the maximum number of in flight messages is reached and `AddPermission` returns this error if the maximum number of permissions for the queue is reached.

HTTP Status Code: 400

QueueDoesNotExist

Ensure that the `QueueUrl` is correct and that the queue has not been deleted.

HTTP Status Code: 400

RequestThrottled

The request was denied due to request throttling.

- Exceeds the permitted request rate for the queue or for the recipient of the request.
- Ensure that the request rate is within the Amazon SQS limits for sending messages. For more information, see Amazon SQS quotas in the *Amazon SQS Developer Guide*.

HTTP Status Code: 400

UnsupportedOperation

Error code 400. Unsupported operation.

HTTP Status Code: 400

Examples

The following example query request receives messages from the specified queue. The structure of `AUTHPARAMS` depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the *AWS General Reference*.

Example

Using AWS JSON protocol (Default)

Sample Request

```
POST / HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Target: AmazonSQS.ReceiveMessage
X-Amz-Date: <Date>
Content-Type: application/x-amz-json-1.0
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
{
    "QueueUrl": "https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue/",
    "MaxNumberOfMessages": 5,
    "VisibilityTimeout": 15,
    "AttributeNames": ["All"]
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <requestId>
Content-Length: <PayloadSizeBytes>
Date: <Date>
Content-Type: application/x-amz-json-1.0
{
    "Messages": [
        {
            "Attributes": {
                "SenderId": "AIDASSYFHUBOBT7F4XT75",
                "ApproximateFirstReceiveTimestamp": "1677112433437",
                "ApproximateReceiveCount": "1",
                "SentTimestamp": "1677112427387"
            },
            "Body": "This is a test message",
            "MD5OfBody": "fafb00f5732ab283681e124bf8747ed1",
            "MessageId": "219f8380-5770-4cc2-8c3e-5c715e145f5e",
            "ReceiptHandle": "AQEBaZ+j5qUoOAoxlmrCQPkBm9njMWXqemmIG6shMHCO6fV20JrQYg/
AiZ8JELwLwOu5U61W+aIX5Qzu7GGofxJuvzymr4Ph53RiR0mudj4InLSgpSspYeTRDteBye5tV/txbZDdNZxsi
+qqZA9xPnmMscKQqF6pGhnGIKrnkYGl45Nl6GPIZv62LrIRb6mSqOn1fn0yqrvmWuuY3w2UzQbaYunJWGxpzZze21EOBtyw
```

```
g7G9is
+c6K9hGniddzhLkK1tHzZKjejOU4jokaiB4nmi0dF3JqLzDsQuPF0Gi8qffhEvw56nl8QCbluSJScFhJYvoagGnDbwOnd9z
NGjWJ45dqWpiW3d2Ws7q"
        }
    ]
}
```

Example

Using AWS query protocol

Sample Request

```
POST /177715257436/MyQueue/ HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Date: <Date>
Content-Type: application/x-www-form-urlencoded
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
Action=ReceiveMessage
&MaxNumberOfMessages=5
&VisibilityTimeout=15
&AttributeName=All
```

Sample Response

```
HTTP/1.1 200 OK
<ReceiveMessageResponse xmlns="http://queue.amazonaws.com/doc/2012-11-05/">
    <ReceiveMessageResult>
        <Message>
            <MessageId>60e827c3-c8a5-410a-af0e-fb43746e70b1</MessageId>
            <ReceiptHandle>AQEBwPTK2fT2gy97H1iyU5in9umgT+Y4IOxyKGOzpZa8iemEqoR5/
aPn0xAodmiVTzyrW7S4e8XwcWbB04XK92jIQzUpiGwRFA4Dl7r3GOw84Qzq/0OBQe/
JaKxJw6iilafYA5fo1SJQo5Wg8xXbJHTVlJqgvTXd/
UtlByLMhWMi0JMra1UUjYiPsGtYUpLVnOaRkYSPvzRnFFYUbcqCW9lm2Bi/
jQKK6KNOZyCCfIh8TooE5i4P2L9N3o9yUHwMdv6p0nb5lKaGurQ2sJwwsyhXf38ZHnVN6pWwsqQnWKYuEXpxPofxd2lcLdg
ReceiptHandle>
            <MD5OfBody>0e024d309850c78cba5eabbeff7cae71</MD5OfBody>
            <Body>test message body 1</Body>
            <Attribute>
                <Name>SenderId</Name>
                <Value>AIDASSYFHUBOBT7F4XT75</Value>
```

```
            </Attribute>
            <Attribute>
                <Name>ApproximateFirstReceiveTimestamp</Name>
                <Value>1677112300463</Value>
            </Attribute>
            <Attribute>
                <Name>ApproximateReceiveCount</Name>
                <Value>1</Value>
            </Attribute>
            <Attribute>
                <Name>SentTimestamp</Name>
                <Value>1677111805489</Value>
            </Attribute>
        </Message>
    </ReceiveMessageResult>
    <ResponseMetadata>
        <RequestId>5ba605cc-1e4b-58ba-93db-59bca8677ec9</RequestId>
    </ResponseMetadata>
</ReceiveMessageResponse>
```

Example

The following example enables long polling by calling the `ReceiveMessage` action with the `WaitTimeSeconds` parameter set to 10 seconds.

Using AWS JSON protocol (Default)

Sample Request

```
POST / HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Target: AmazonSQS.ReceiveMessage
X-Amz-Date: <Date>
Content-Type: application/x-amz-json-1.0
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
{
    "QueueUrl": "https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue/",
    "WaitTimeSeconds": 10,
    "MaxNumberOfMessages": 5,
    "VisibilityTimeout": 15,
    "AttributeNames": ["All"]
```

```
}
```

Example

The following example shows the request and response when using the parameter `MessageSystemAttributeNames`.

Sample Request

```
aws sqs receive-message \
  --queue-url https://sqs.us-east-1.amazonaws.com/123456789012/MyQueue \
  --message-system-attribute-names SentTimestamp SenderId
```

Sample Response

```
{
  "Messages": [
    {
      "MessageId": "abc1234d-5678-90ab-cdef-EXAMPLE11111",
      "ReceiptHandle": "AQEBwJnKyrHigUMZj6rYigCgxlaS3SLy0a...",
      "MD5OfBody": "e99a18c428cb38d5f260853678922e03",
      "Body": "Example message",
      "Attributes": {
        "SenderId": "AIDAEXAMPLE123ABC",
        "SentTimestamp": "1638368280000"
      }
    }
  ]
}
```

Example

Using AWS query protocol

Sample Request

```
POST /177715257436/MyQueue/ HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Date: <Date>
Content-Type: application/x-www-form-urlencoded
```

```
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
Action=ReceiveMessage
&WaitTimeSeconds=10
&MaxNumberOfMessages=5
&VisibilityTimeout=15
&AttributeName=All
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python (Boto3)
- AWS SDK for Ruby V3

RemovePermission

Revokes any permissions in the queue policy that matches the specified `Label` parameter.

Note

- Only the owner of a queue can remove permissions from it.
- Cross-account permissions don't apply to this action. For more information, see Grant cross-account permissions to a role and a username in the *Amazon SQS Developer Guide*.
- To remove the ability to change queue permissions, you must deny permission to the `AddPermission`, `RemovePermission`, and `SetQueueAttributes` actions in your IAM policy.

Request Syntax

```
{
   "Label": "string",
   "QueueUrl": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

Label

The identification of the permission to remove. This is the label added using the `AddPermission` action.

Type: String

Required: Yes

QueueUrl

The URL of the Amazon SQS queue from which permissions are removed.

Queue URLs and names are case-sensitive.

Type: String

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors.

InvalidAddress

The specified ID is invalid.

HTTP Status Code: 400

InvalidSecurity

The request was not made over HTTPS or did not use SigV4 for signing.

HTTP Status Code: 400

QueueDoesNotExist

Ensure that the `QueueUrl` is correct and that the queue has not been deleted.

HTTP Status Code: 400

RequestThrottled

The request was denied due to request throttling.

- Exceeds the permitted request rate for the queue or for the recipient of the request.
- Ensure that the request rate is within the Amazon SQS limits for sending messages. For more information, see Amazon SQS quotas in the *Amazon SQS Developer Guide*.

HTTP Status Code: 400

UnsupportedOperation

Error code 400. Unsupported operation.

HTTP Status Code: 400

Examples

The following example query request removes the `MyLabel` permission from the queue named `MyQueue`. The structure of `AUTHPARAMS` depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the *AWS General Reference*.

Example

Using AWS JSON protocol (Default)

Sample Request

```
POST / HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Target: AmazonSQS.RemovePermission
X-Amz-Date: <Date>
Content-Type: application/x-amz-json-1.0
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
{
    "QueueUrl": "https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue/",
    "Label": "MyLabel"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <requestId>
Content-Length: 0
Date: <Date>
Content-Type: application/x-amz-json-1.0
```

Example

Using AWS query protocol

Sample Request

```
POST /177715257436/MyQueue/ HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
```

```
X-Amz-Date: <Date>
Content-Type: application/x-www-form-urlencoded
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
Action=RemovePermission&Label=MyLabel
```

Sample Response

```
HTTP/1.1 200 OK
<?xml version="1.0"?>
<RemovePermissionResponse xmlns="http://queue.amazonaws.com/doc/2012-11-05/">
    <ResponseMetadata>
        <RequestId>133cc8b8-21a1-5aba-b65a-94c8e50d917f</RequestId>
    </ResponseMetadata>
</RemovePermissionResponse>
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python (Boto3)
- AWS SDK for Ruby V3

SendMessage

Delivers a message to the specified queue.

> **Important**
>
> A message can include only XML, JSON, and unformatted text. The following Unicode characters are allowed. For more information, see the W3C specification for characters.
> #x9 | #xA | #xD | #x20 to #xD7FF | #xE000 to #xFFFD | #x10000 to #x10FFFF
> If a message contains characters outside the allowed set, Amazon SQS rejects the message and returns an InvalidMessageContents error. Ensure that your message body includes only valid characters to avoid this exception.

Request Syntax

```
{
   "DelaySeconds": number,
   "MessageAttributes": {
      "string" : {
         "BinaryListValues": [ blob ],
         "BinaryValue": blob,
         "DataType": "string",
         "StringListValues": [ "string" ],
         "StringValue": "string"
      }
   },
   "MessageBody": "string",
   "MessageDeduplicationId": "string",
   "MessageGroupId": "string",
   "MessageSystemAttributes": {
      "string" : {
         "BinaryListValues": [ blob ],
         "BinaryValue": blob,
         "DataType": "string",
         "StringListValues": [ "string" ],
         "StringValue": "string"
      }
   },
   "QueueUrl": "string"
```

```
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

DelaySeconds

The length of time, in seconds, for which to delay a specific message. Valid values: 0 to 900. Maximum: 15 minutes. Messages with a positive `DelaySeconds` value become available for processing after the delay period is finished. If you don't specify a value, the default value for the queue applies.

> **Note**
>
> When you set `FifoQueue`, you can't set `DelaySeconds` per message. You can set this parameter only on a queue level.

Type: Integer

Required: No

MessageAttributes

Each message attribute consists of a `Name`, `Type`, and `Value`. For more information, see Amazon SQS message attributes in the *Amazon SQS Developer Guide*.

Type: String to MessageAttributeValue object map

Required: No

MessageBody

The message to send. The minimum size is one character. The maximum size is 1 MiB or 1,048,576 bytes

> **Important**
>
> A message can include only XML, JSON, and unformatted text. The following Unicode characters are allowed. For more information, see the W3C specification for characters.

#x9 | #xA | #xD | #x20 to #xD7FF | #xE000 to #xFFFD | #x10000 to #x10FFFF
If a message contains characters outside the allowed set, Amazon SQS rejects the message and returns an InvalidMessageContents error. Ensure that your message body includes only valid characters to avoid this exception.

Type: String

Required: Yes

MessageDeduplicationId

This parameter applies only to FIFO (first-in-first-out) queues.

The token used for deduplication of sent messages. If a message with a particular `MessageDeduplicationId` is sent successfully, any messages sent with the same `MessageDeduplicationId` are accepted successfully but aren't delivered during the 5-minute deduplication interval. For more information, see Exactly-once processing in the *Amazon SQS Developer Guide*.

- Every message must have a unique `MessageDeduplicationId`,
  - You may provide a `MessageDeduplicationId` explicitly.
  - If you aren't able to provide a `MessageDeduplicationId` and you enable `ContentBasedDeduplication` for your queue, Amazon SQS uses a SHA-256 hash to generate the `MessageDeduplicationId` using the body of the message (but not the attributes of the message).
  - If you don't provide a `MessageDeduplicationId` and the queue doesn't have `ContentBasedDeduplication` set, the action fails with an error.
  - If the queue has `ContentBasedDeduplication` set, your `MessageDeduplicationId` overrides the generated one.
- When `ContentBasedDeduplication` is in effect, messages with identical content sent within the deduplication interval are treated as duplicates and only one copy of the message is delivered.
- If you send one message with `ContentBasedDeduplication` enabled and then another message with a `MessageDeduplicationId` that is the same as the one generated for the first `MessageDeduplicationId`, the two messages are treated as duplicates and only one copy of the message is delivered.

> **Note**
>
> The `MessageDeduplicationId` is available to the consumer of the message (this can be useful for troubleshooting delivery issues).
> If a message is sent successfully but the acknowledgement is lost and the message is resent with the same `MessageDeduplicationId` after the deduplication interval, Amazon SQS can't detect duplicate messages.
> Amazon SQS continues to keep track of the message deduplication ID even after the message is received and deleted.

The maximum length of `MessageDeduplicationId` is 128 characters. `MessageDeduplicationId` can contain alphanumeric characters (`a-z`, `A-Z`, `0-9`) and punctuation (`!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~`).

For best practices of using `MessageDeduplicationId`, see Using the MessageDeduplicationId Property in the *Amazon SQS Developer Guide*.

Type: String

Required: No

MessageGroupId

`MessageGroupId` is an attribute used in Amazon SQS FIFO (First-In-First-Out) and standard queues. In FIFO queues, `MessageGroupId` organizes messages into distinct groups. Messages within the same message group are always processed one at a time, in strict order, ensuring that no two messages from the same group are processed simultaneously. In standard queues, using `MessageGroupId` enables fair queues. It is used to identify the tenant a message belongs to, helping maintain consistent message dwell time across all tenants during noisy neighbor events. Unlike FIFO queues, messages with the same `MessageGroupId` can be processed in parallel, maintaining the high throughput of standard queues.

- **FIFO queues:** `MessageGroupId` acts as the tag that specifies that a message belongs to a specific message group. Messages that belong to the same message group are processed in a FIFO manner (however, messages in different message groups might be processed out of order). To interleave multiple ordered streams within a single queue, use `MessageGroupId` values (for example, session data for multiple users). In this scenario, multiple consumers can process the queue, but the session data of each user is processed in a FIFO fashion.

If you do not provide a `MessageGroupId` when sending a message to a FIFO queue, the action fails.

`ReceiveMessage` might return messages with multiple `MessageGroupId` values. For each `MessageGroupId`, the messages are sorted by time sent.

- **Standard queues:**Use `MessageGroupId` in standard queues to enable fair queues. The `MessageGroupId` identifies the tenant a message belongs to. A tenant can be any entity that shares a queue with others, such as your customer, a client application, or a request type. When one tenant sends a disproportionately large volume of messages or has messages that require longer processing time, fair queues ensure other tenants' messages maintain low dwell time. This preserves quality of service for all tenants while maintaining the scalability and throughput of standard queues. We recommend that you include a `MessageGroupId` in all messages when using fair queues.

The length of `MessageGroupId` is 128 characters. Valid values: alphanumeric characters and punctuation `(!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~)`.

For best practices of using `MessageGroupId`, see Using the MessageGroupId Property in the *Amazon SQS Developer Guide*.

Type: String

Required: No

MessageSystemAttributes

The message system attribute to send. Each message system attribute consists of a `Name`, `Type`, and `Value`.

> **Important**
>
> - Currently, the only supported message system attribute is `AWSTraceHeader`. Its type must be `String` and its value must be a correctly formatted AWS X-Ray trace header string.
> - The size of a message system attribute doesn't count towards the total size of a message.

Type: String to MessageSystemAttributeValue object map

Valid Keys: `AWSTraceHeader`

Required: No

QueueUrl

The URL of the Amazon SQS queue to which a message is sent.

Queue URLs and names are case-sensitive.

Type: String

Required: Yes

Response Syntax

```
{
   "MD5OfMessageAttributes": "string",
   "MD5OfMessageBody": "string",
   "MD5OfMessageSystemAttributes": "string",
   "MessageId": "string",
   "SequenceNumber": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

MD5OfMessageAttributes

An MD5 digest of the non-URL-encoded message attribute string. You can use this attribute to verify that Amazon SQS received the message correctly. Amazon SQS URL-decodes the message before creating the MD5 digest. For information about MD5, see RFC1321.

Type: String

MD5OfMessageBody

An MD5 digest of the non-URL-encoded message body string. You can use this attribute to verify that Amazon SQS received the message correctly. Amazon SQS URL-decodes the message before creating the MD5 digest. For information about MD5, see RFC1321.

Type: String

MD5OfMessageSystemAttributes

An MD5 digest of the non-URL-encoded message system attribute string. You can use this attribute to verify that Amazon SQS received the message correctly. Amazon SQS URL-decodes the message before creating the MD5 digest.

Type: String

MessageId

An attribute containing the `MessageId` of the message sent to the queue. For more information, see Queue and Message Identifiers in the *Amazon SQS Developer Guide*.

Type: String

SequenceNumber

This parameter applies only to FIFO (first-in-first-out) queues.

The large, non-consecutive number that Amazon SQS assigns to each message.

The length of `SequenceNumber` is 128 bits. `SequenceNumber` continues to increase for a particular `MessageGroupId`.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

InvalidAddress

The specified ID is invalid.

HTTP Status Code: 400

InvalidMessageContents

The message contains characters outside the allowed set.

HTTP Status Code: 400

InvalidSecurity

The request was not made over HTTPS or did not use SigV4 for signing.

HTTP Status Code: 400

KmsAccessDenied

The caller doesn't have the required KMS access.

HTTP Status Code: 400

KmsDisabled

The request was denied due to request throttling.

HTTP Status Code: 400

KmsInvalidKeyUsage

The request was rejected for one of the following reasons:

- The KeyUsage value of the KMS key is incompatible with the API operation.
- The encryption algorithm or signing algorithm specified for the operation is incompatible with the type of key material in the KMS key (KeySpec).

HTTP Status Code: 400

KmsInvalidState

The request was rejected because the state of the specified resource is not valid for this request.

HTTP Status Code: 400

KmsNotFound

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 400

KmsOptInRequired

The request was rejected because the specified key policy isn't syntactically or semantically correct.

HTTP Status Code: 400

KmsThrottled

AWS KMS throttles requests for the following conditions.

HTTP Status Code: 400

QueueDoesNotExist

Ensure that the `QueueUrl` is correct and that the queue has not been deleted.

HTTP Status Code: 400

RequestThrottled

The request was denied due to request throttling.

- Exceeds the permitted request rate for the queue or for the recipient of the request.
- Ensure that the request rate is within the Amazon SQS limits for sending messages. For more information, see Amazon SQS quotas in the *Amazon SQS Developer Guide*.

HTTP Status Code: 400

UnsupportedOperation

Error code 400. Unsupported operation.

HTTP Status Code: 400

Examples

The following example `SendMessage` request sends a message containing `This is a test message` to the queue. You must URL-encode the entire URL. However, in this example only the message body is URL-encoded to make the example easier to read. The structure of `AUTHPARAMS` depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the *AWS General Reference*.

Example

Using AWS JSON protocol (Default)

Sample Request

```
POST / HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
```

```
X-Amz-Target: AmazonSQS.SendMessage
X-Amz-Date: <Date>
Content-Type: application/x-amz-json-1.0
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
{
    "QueueUrl": "https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue/",
    "MessageBody": "This is a test message",
    "MessageAttributes": {
        "my_attribute_name_1": {
            "DataType": "String",
            "StringValue": "my_attribute_value_1"
        },
        "my_attribute_name_2": {
            "DataType": "String",
            "StringValue": "my_attribute_value_2"
        }
    }
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <requestId>
Content-Length: <PayloadSizeBytes>
Date: <Date>
Content-Type: application/x-amz-json-1.0
{
    "MD5OfMessageAttributes": "c48838208d2b4e14e3ca0093a8443f09",
    "MD5OfMessageBody": "fafb00f5732ab283681e124bf8747ed1",
    "MessageId": "219f8380-5770-4cc2-8c3e-5c715e145f5e"
}
```

Example

Using AWS query protocol

Sample Request

```
POST /177715257436/MyQueue/ HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Date: <Date>
```

```
Content-Type: application/x-www-form-urlencoded
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
Action=SendMessage
&MessageBody=This+is+a+test+message
&MessageAttribute.1.Name=my_attribute_name_1
&MessageAttribute.1.Value.StringValue=my_attribute_value_1
&MessageAttribute.1.Value.DataType=String
&MessageAttribute.2.Name=my_attribute_name_2
&MessageAttribute.2.Value.StringValue=my_attribute_value_2
&MessageAttribute.2.Value.DataType=String
```

Sample Response

```
HTTP/1.1 200 OK
<?xml version="1.0"?>
<SendMessageResponse xmlns="http://queue.amazonaws.com/doc/2012-11-05/">
    <SendMessageResult>
        <MessageId>374cec7b-d0c8-4a2e-ad0b-67be763cf97e</MessageId>
        <MD5OfMessageBody>fafb00f5732ab283681e124bf8747ed1</MD5OfMessageBody>
        <MD5OfMessageAttributes>c48838208d2b4e14e3ca0093a8443f09</
MD5OfMessageAttributes>
    </SendMessageResult>
    <ResponseMetadata>
        <RequestId>7fe4446e-b452-53f7-8f85-181e06f2dd99</RequestId>
    </ResponseMetadata>
</SendMessageResponse>
```

Example

The following example creates a *message timer*—applying a 45-second initial visibility delay to a single message— by calling the `SendMessage` action with the `DelaySeconds` parameter set to 45 seconds.

> **Note**
>
> Queue URLs and names are case-sensitive.

Using AWS JSON protocol (Default)

Sample Request

```
POST / HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Target: AmazonSQS.SendMessage
X-Amz-Date: <Date>
Content-Type: application/x-amz-json-1.0
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
{
    "QueueUrl": "https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue/",
    "MessageBody": "This is a test message",
    "DelaySeconds": 45
}
```

Example

Using AWS query protocol

Sample Request

```
POST /177715257436/MyQueue/ HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Date: <Date>
Content-Type: application/x-www-form-urlencoded
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
Action=SendMessage
&MessageBody=This+is+a+test+message
&DelaySeconds=45
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++

- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python (Boto3)
- AWS SDK for Ruby V3

SendMessageBatch

You can use `SendMessageBatch` to send up to 10 messages to the specified queue by assigning either identical or different values to each message (or by not assigning values at all). This is a batch version of `SendMessage`. For a FIFO queue, multiple messages within a single batch are enqueued in the order they are sent.

The result of sending each message is reported individually in the response. Because the batch request can result in a combination of successful and unsuccessful actions, you should check for batch errors even when the call returns an HTTP status code of `200`.

The maximum allowed individual message size and the maximum total payload size (the sum of the individual lengths of all of the batched messages) are both 1 MiB 1,048,576 bytes.

> **Important**
>
> A message can include only XML, JSON, and unformatted text. The following Unicode characters are allowed. For more information, see the W3C specification for characters.
> `#x9` | `#xA` | `#xD` | `#x20` to `#xD7FF` | `#xE000` to `#xFFFD` | `#x10000` to `#x10FFFF`
> If a message contains characters outside the allowed set, Amazon SQS rejects the message and returns an InvalidMessageContents error. Ensure that your message body includes only valid characters to avoid this exception.

If you don't specify the `DelaySeconds` parameter for an entry, Amazon SQS uses the default value for the queue.

Request Syntax

```
{
   "Entries": [
      {
         "DelaySeconds": number,
         "Id": "string",
         "MessageAttributes": {
            "string" : {
               "BinaryListValues": [ blob ],
               "BinaryValue": blob,
               "DataType": "string",
```

```
                "StringListValues": [ "string" ],
                "StringValue": "string"
             }
          },
          "MessageBody": "string",
          "MessageDeduplicationId": "string",
          "MessageGroupId": "string",
          "MessageSystemAttributes": {
             "string" : {
                "BinaryListValues": [ blob ],
                "BinaryValue": blob,
                "DataType": "string",
                "StringListValues": [ "string" ],
                "StringValue": "string"
             }
          }
       }
    ],
    "QueueUrl": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

Entries

A list of SendMessageBatchRequestEntry items.

Type: Array of SendMessageBatchRequestEntry objects

Required: Yes

QueueUrl

The URL of the Amazon SQS queue to which batched messages are sent.

Queue URLs and names are case-sensitive.

Type: String

Required: Yes

Response Syntax

```
{
   "Failed": [
      {
         "Code": "string",
         "Id": "string",
         "Message": "string",
         "SenderFault": boolean
      }
   ],
   "Successful": [
      {
         "Id": "string",
         "MD5OfMessageAttributes": "string",
         "MD5OfMessageBody": "string",
         "MD5OfMessageSystemAttributes": "string",
         "MessageId": "string",
         "SequenceNumber": "string"
      }
   ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Failed

A list of BatchResultErrorEntry items with error details about each message that can't be enqueued.

Type: Array of BatchResultErrorEntry objects

Successful

A list of SendMessageBatchResultEntry items.

Type: Array of SendMessageBatchResultEntry objects

Errors

For information about the errors that are common to all actions, see Common Errors.

BatchEntryIdsNotDistinct

Two or more batch entries in the request have the same `Id`.

HTTP Status Code: 400

BatchRequestTooLong

The length of all the messages put together is more than the limit.

HTTP Status Code: 400

EmptyBatchRequest

The batch request doesn't contain any entries.

HTTP Status Code: 400

InvalidAddress

The specified ID is invalid.

HTTP Status Code: 400

InvalidBatchEntryId

The `Id` of a batch entry in a batch request doesn't abide by the specification.

HTTP Status Code: 400

InvalidSecurity

The request was not made over HTTPS or did not use SigV4 for signing.

HTTP Status Code: 400

KmsAccessDenied

The caller doesn't have the required KMS access.

HTTP Status Code: 400

KmsDisabled

The request was denied due to request throttling.

HTTP Status Code: 400

KmsInvalidKeyUsage

The request was rejected for one of the following reasons:

- The KeyUsage value of the KMS key is incompatible with the API operation.
- The encryption algorithm or signing algorithm specified for the operation is incompatible with the type of key material in the KMS key (KeySpec).

HTTP Status Code: 400

KmsInvalidState

The request was rejected because the state of the specified resource is not valid for this request.

HTTP Status Code: 400

KmsNotFound

The request was rejected because the specified entity or resource could not be found.

HTTP Status Code: 400

KmsOptInRequired

The request was rejected because the specified key policy isn't syntactically or semantically correct.

HTTP Status Code: 400

KmsThrottled

AWS KMS throttles requests for the following conditions.

HTTP Status Code: 400

QueueDoesNotExist

Ensure that the `QueueUrl` is correct and that the queue has not been deleted.

HTTP Status Code: 400

RequestThrottled

The request was denied due to request throttling.

- Exceeds the permitted request rate for the queue or for the recipient of the request.

- Ensure that the request rate is within the Amazon SQS limits for sending messages. For more information, see Amazon SQS quotas in the *Amazon SQS Developer Guide*.

HTTP Status Code: 400

TooManyEntriesInBatchRequest

The batch request contains more entries than permissible. For Amazon SQS, the maximum number of entries you can include in a single SendMessageBatch, DeleteMessageBatch, or ChangeMessageVisibilityBatch request is 10.

HTTP Status Code: 400

UnsupportedOperation

Error code 400. Unsupported operation.

HTTP Status Code: 400

Examples

The following example SendMessageBatch request sends two messages to the queue. You must URL-encode the entire URL. However, in this example only the message body is URL-encoded to make the example easier to read. The structure of AUTHPARAMS depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the *AWS General Reference*.

Example

Using AWS JSON protocol (Default)

Sample Request

```
POST / HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Target: AmazonSQS.SendMessageBatch
X-Amz-Date: <Date>
Content-Type: application/x-amz-json-1.0
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
{
```

```
    "QueueUrl": "https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue/",
    "Entries": [
        {
            "Id": "test_msg_001",
            "MessageBody": "test message body 1"
        },
        {
            "Id": "test_msg_002",
            "MessageBody": "test message body 2",
            "DelaySeconds": 60,
            "MessageAttributes": {
                "my_attribute_name_1": {
                    "DataType": "String",
                    "StringValue": "my_attribute_value_1"
                }
            }
        }
    ]
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <requestId>
Content-Length: <PayloadSizeBytes>
Date: <Date>
Content-Type: application/x-amz-json-1.0
{
   "Failed": [],
    "Successful": [
        {
            "Id": "test_msg_001",
            "MD5OfMessageBody": "0e024d309850c78cba5eabbeff7cae71",
            "MessageId": "f4eb349f-cd33-4bc4-bdc2-e557c900d41d"
        },
        {
            "Id": "test_msg_002",
            "MD5OfMessageAttributes": "8ef4d60dbc8efda9f260e1dfd09d29f3",
            "MD5OfMessageBody": "27118326006d3829667a400ad23d5d98",
            "MessageId": "1dcfcd50-5a67-45ae-ae4c-1c152b5effb9"
        }
    ]
}
```

Example

Using AWS query protocol

Sample Request

```
POST /177715257436/MyQueue/ HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Date: <Date>
Content-Type: application/x-www-form-urlencoded
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
Action=SendMessageBatch
&SendMessageBatchRequestEntry.1.Id=test_msg_001
&SendMessageBatchRequestEntry.1.MessageBody=test%20message%20body%201
&SendMessageBatchRequestEntry.2.Id=test_msg_002
&SendMessageBatchRequestEntry.2.MessageBody=test%20message%20body%202
&SendMessageBatchRequestEntry.2.DelaySeconds=60
&SendMessageBatchRequestEntry.2.MessageAttribute.1.Name=test_attribute_name_1
&SendMessageBatchRequestEntry.2.MessageAttribute.1.Value.StringValue=test_attribute_value_1
&SendMessageBatchRequestEntry.2.MessageAttribute.1.Value.DataType=String
```

Sample Response

```
HTTP/1.1 200 OK
<?xml version="1.0"?>
<SendMessageBatchResponse xmlns="http://queue.amazonaws.com/doc/2012-11-05/">
    <SendMessageBatchResult>
        <SendMessageBatchResultEntry>
            <Id>test_msg_001</Id>
            <MessageId>60e827c3-c8a5-410a-af0e-fb43746e70b1</MessageId>
            <MD5OfMessageBody>0e024d309850c78cba5eabbeff7cae71</MD5OfMessageBody>
        </SendMessageBatchResultEntry>
        <SendMessageBatchResultEntry>
            <Id>test_msg_00</Id>
            <MessageId>c6e7fc6a-b802-4724-be06-59833004578b</MessageId>
            <MD5OfMessageBody>7fb8146a82f95e0af155278f406862c2</MD5OfMessageBody>
            <MD5OfMessageAttributes>ba056227cfd9533dba1f72ad9816d233</
MD5OfMessageAttributes>
        </SendMessageBatchResultEntry>
    </SendMessageBatchResult>
    <ResponseMetadata>
```

```
        <RequestId>5150a701-14f7-5609-b136-fb71a0ca744a</RequestId>
    </ResponseMetadata>
</SendMessageBatchResponse>
```

Example

The following example sends multiple messages with *message timers*—applying a visibility delay of variable length to the messages in the batch—by calling the `SendMessageBatch` action *without* a value for `DelaySeconds` for the first message and with the values of 45 seconds and 2 minutes for the second and third messages.

> **Note**
>
> If you don't set a value for the `DelaySeconds` parameter, the message might still be subject to a delay if you add the message to a *delay queue*. For more information about using delay queues, see Amazon SQS Delay Queues in the *Amazon SQS Developer Guide*.

Using AWS JSON protocol (Default)

Sample Request

```
POST / HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Target: AmazonSQS.SendMessageBatch
X-Amz-Date: <Date>
Content-Type: application/x-amz-json-1.0
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
{
    "QueueUrl": "https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue/",
    "Entries": [
        {
            "Id": "test_msg_no_message_timer",
            "MessageBody": "test message body 1"
        },
        {
            "Id": "test_msg_delay_45_seconds",
            "MessageBody": "test message body 2",
            "DelaySeconds": 45
```

```
        },
        {
            "Id": "test_msg_delay_2_minutes",
            "MessageBody": "test message body 3",
            "DelaySeconds": 120
        }
    ]
}
```

Example

Using AWS query protocol

Sample Request

```
POST /177715257436/MyQueue/ HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Date: <Date>
Content-Type: application/x-www-form-urlencoded
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
Action=SendMessageBatch
&SendMessageBatchRequestEntry.1.Id=test_msg_no_message_timer
&SendMessageBatchRequestEntry.1.MessageBody=test%20message%20body%201
&SendMessageBatchRequestEntry.2.Id=test_msg_delay_45_seconds
&SendMessageBatchRequestEntry.2.MessageBody=test%20message%20body%202
&SendMessageBatchRequestEntry.2.DelaySeconds=45
&SendMessageBatchRequestEntry.3.Id=test_msg_delay_2_minutes
&SendMessageBatchRequestEntry.3.MessageBody=test%20message%20body%203
&SendMessageBatchRequestEntry.3.DelaySeconds=120
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2

- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python (Boto3)
- AWS SDK for Ruby V3

SetQueueAttributes

Sets the value of one or more queue attributes, like a policy. When you change a queue's attributes, the change can take up to 60 seconds for most of the attributes to propagate throughout the Amazon SQS system. Changes made to the `MessageRetentionPeriod` attribute can take up to 15 minutes and will impact existing messages in the queue potentially causing them to be expired and deleted if the `MessageRetentionPeriod` is reduced below the age of existing messages.

Note

- In the future, new attributes might be added. If you write code that calls this action, we recommend that you structure your code so that it can handle new attributes gracefully.
- Cross-account permissions don't apply to this action. For more information, see Grant cross-account permissions to a role and a username in the *Amazon SQS Developer Guide*.
- To remove the ability to change queue permissions, you must deny permission to the `AddPermission`, `RemovePermission`, and `SetQueueAttributes` actions in your IAM policy.

Request Syntax

```
{
   "Attributes": {
      "string" : "string"
   },
   "QueueUrl": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

Attributes

A map of attributes to set.

The following lists the names, descriptions, and values of the special request parameters that the `SetQueueAttributes` action uses:

- `DelaySeconds` – The length of time, in seconds, for which the delivery of all messages in the queue is delayed. Valid values: An integer from 0 to 900 (15 minutes). Default: 0.
- `MaximumMessageSize` – The limit of how many bytes a message can contain before Amazon SQS rejects it. Valid values: An integer from 1,024 bytes (1 KiB) up to 1,048,576 bytes (1 MiB). Default: 1,048,576 bytes (1 MiB).
- `MessageRetentionPeriod` – The length of time, in seconds, for which Amazon SQS retains a message. Valid values: An integer representing seconds, from 60 (1 minute) to 1,209,600 (14 days). Default: 345,600 (4 days). When you change a queue's attributes, the change can take up to 60 seconds for most of the attributes to propagate throughout the Amazon SQS system. Changes made to the `MessageRetentionPeriod` attribute can take up to 15 minutes and will impact existing messages in the queue potentially causing them to be expired and deleted if the `MessageRetentionPeriod` is reduced below the age of existing messages.
- `Policy` – The queue's policy. A valid AWS policy. For more information about policy structure, see Overview of AWS IAM Policies in the *AWS Identity and Access Management User Guide*.
- `ReceiveMessageWaitTimeSeconds` – The length of time, in seconds, for which a `ReceiveMessage` action waits for a message to arrive. Valid values: An integer from 0 to 20 (seconds). Default: 0.
- `VisibilityTimeout` – The visibility timeout for the queue, in seconds. Valid values: An integer from 0 to 43,200 (12 hours). Default: 30. For more information about the visibility timeout, see Visibility Timeout in the *Amazon SQS Developer Guide*.

The following attributes apply only to dead-letter queues:

- `RedrivePolicy` – The string that includes the parameters for the dead-letter queue functionality of the source queue as a JSON object. The parameters are as follows:
  - `deadLetterTargetArn` – The Amazon Resource Name (ARN) of the dead-letter queue to which Amazon SQS moves messages after the value of `maxReceiveCount` is exceeded.
  - `maxReceiveCount` – The number of times a message is delivered to the source queue before being moved to the dead-letter queue. Default: 10. When the `ReceiveCount` for a message exceeds the `maxReceiveCount` for a queue, Amazon SQS moves the message to the dead-letter-queue.

- `RedriveAllowPolicy` – The string that includes the parameters for the permissions for the dead-letter queue redrive permission and which source queues can specify dead-letter queues as a JSON object. The parameters are as follows:
  - `redrivePermission` – The permission type that defines which source queues can specify the current queue as the dead-letter queue. Valid values are:
    - `allowAll` – (Default) Any source queues in this AWS account in the same Region can specify this queue as the dead-letter queue.
    - `denyAll` – No source queues can specify this queue as the dead-letter queue.
    - `byQueue` – Only queues specified by the `sourceQueueArns` parameter can specify this queue as the dead-letter queue.
  - `sourceQueueArns` – The Amazon Resource Names (ARN)s of the source queues that can specify this queue as the dead-letter queue and redrive messages. You can specify this parameter only when the `redrivePermission` parameter is set to `byQueue`. You can specify up to 10 source queue ARNs. To allow more than 10 source queues to specify dead-letter queues, set the `redrivePermission` parameter to `allowAll`.

> **Note**
>
> The dead-letter queue of a FIFO queue must also be a FIFO queue. Similarly, the dead-letter queue of a standard queue must also be a standard queue.

The following attributes apply only to server-side-encryption:

- `KmsMasterKeyId` – The ID of an AWS managed customer master key (CMK) for Amazon SQS or a custom CMK. For more information, see Key Terms. While the alias of the AWS-managed CMK for Amazon SQS is always `alias/aws/sqs`, the alias of a custom CMK can, for example, be `alias/MyAlias` . For more examples, see KeyId in the *AWS Key Management Service API Reference*.
- `KmsDataKeyReusePeriodSeconds` – The length of time, in seconds, for which Amazon SQS can reuse a data key to encrypt or decrypt messages before calling AWS KMS again. An integer representing seconds, between 60 seconds (1 minute) and 86,400 seconds (24 hours). Default: 300 (5 minutes). A shorter time period provides better security but results in more calls to KMS which might incur charges after Free Tier. For more information, see How Does the Data Key Reuse Period Work?.

- `SqsManagedSseEnabled` – Enables server-side queue encryption using SQS owned encryption keys. Only one server-side encryption option is supported per queue (for example, SSE-KMS or SSE-SQS).

The following attribute applies only to FIFO (first-in-first-out) queues:

- `ContentBasedDeduplication` – Enables content-based deduplication. For more information, see Exactly-once processing in the *Amazon SQS Developer Guide*. Note the following:
  - Every message must have a unique `MessageDeduplicationId`.
    - You may provide a `MessageDeduplicationId` explicitly.
    - If you aren't able to provide a `MessageDeduplicationId` and you enable `ContentBasedDeduplication` for your queue, Amazon SQS uses a SHA-256 hash to generate the `MessageDeduplicationId` using the body of the message (but not the attributes of the message).
    - If you don't provide a `MessageDeduplicationId` and the queue doesn't have `ContentBasedDeduplication` set, the action fails with an error.
    - If the queue has `ContentBasedDeduplication` set, your `MessageDeduplicationId` overrides the generated one.
  - When `ContentBasedDeduplication` is in effect, messages with identical content sent within the deduplication interval are treated as duplicates and only one copy of the message is delivered.
  - If you send one message with `ContentBasedDeduplication` enabled and then another message with a `MessageDeduplicationId` that is the same as the one generated for the first `MessageDeduplicationId`, the two messages are treated as duplicates and only one copy of the message is delivered.

The following attributes apply only to high throughput for FIFO queues:

- `DeduplicationScope` – Specifies whether message deduplication occurs at the message group or queue level. Valid values are `messageGroup` and `queue`.
- `FifoThroughputLimit` – Specifies whether the FIFO queue throughput quota applies to the entire queue or per message group. Valid values are `perQueue` and `perMessageGroupId`. The `perMessageGroupId` value is allowed only when the value for `DeduplicationScope` is `messageGroup`.

To enable high throughput for FIFO queues, do the following:

- Set `DeduplicationScope` to `messageGroup`.
- Set `FifoThroughputLimit` to `perMessageGroupId`.

If you set these attributes to anything other than the values shown for enabling high throughput, normal throughput is in effect and deduplication occurs as specified.

For information on throughput quotas, see Quotas related to messages in the *Amazon SQS Developer Guide*.

Type: String to string map

Valid Keys: `All | Policy | VisibilityTimeout | MaximumMessageSize | MessageRetentionPeriod | ApproximateNumberOfMessages | ApproximateNumberOfMessagesNotVisible | CreatedTimestamp | LastModifiedTimestamp | QueueArn | ApproximateNumberOfMessagesDelayed | DelaySeconds | ReceiveMessageWaitTimeSeconds | RedrivePolicy | FifoQueue | ContentBasedDeduplication | KmsMasterKeyId | KmsDataKeyReusePeriodSeconds | DeduplicationScope | FifoThroughputLimit | RedriveAllowPolicy | SqsManagedSseEnabled`

Required: Yes

QueueUrl

The URL of the Amazon SQS queue whose attributes are set.

Queue URLs and names are case-sensitive.

Type: String

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors.

InvalidAddress

The specified ID is invalid.

HTTP Status Code: 400

InvalidAttributeName

The specified attribute doesn't exist.

HTTP Status Code: 400

InvalidAttributeValue

A queue attribute value is invalid.

HTTP Status Code: 400

InvalidSecurity

The request was not made over HTTPS or did not use SigV4 for signing.

HTTP Status Code: 400

OverLimit

The specified action violates a limit. For example, `ReceiveMessage` returns this error if the maximum number of in flight messages is reached and `AddPermission` returns this error if the maximum number of permissions for the queue is reached.

HTTP Status Code: 400

QueueDoesNotExist

Ensure that the `QueueUrl` is correct and that the queue has not been deleted.

HTTP Status Code: 400

RequestThrottled

The request was denied due to request throttling.

- Exceeds the permitted request rate for the queue or for the recipient of the request.
- Ensure that the request rate is within the Amazon SQS limits for sending messages. For more information, see Amazon SQS quotas in the *Amazon SQS Developer Guide*.

HTTP Status Code: 400

UnsupportedOperation

Error code 400. Unsupported operation.

HTTP Status Code: 400

Examples

The following example query request sets a policy that gives all users `ReceiveMessage` permission for a queue named `MyQueue`. For more examples of policies, see Custom Amazon SQS Access Policy Language Examples in the *Amazon SQS Developer Guide*. The structure of `AUTHPARAMS` depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the *AWS General Reference*.

Example

Using AWS JSON protocol (Default)

Sample Request

```
POST / HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Target: AmazonSQS.SetQueueAtrributes
X-Amz-Date: <Date>
Content-Type: application/x-amz-json-1.0
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive {
    "QueueUrl": "https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue/",
    "Attributes": {
        "Policy": "{\"Version\":\"2012-10-17\",\"Id\":\"Policy1677095510157\",
\"Statement\":[{\"Sid\":\"Stmt1677095506939\",\"Effect\":\"Allow\",\"Principal
\":\"*\",\"Action\":\"sqs:ReceiveMessage\",\"Resource\":\"arn:aws:sqs:us-
east-1:555555555555:MyQueue6\"}]}"
    }
}
```

Sample Response

```
HTTP/1.1 200 OK
```

```
x-amzn-RequestId: <requestId>
Content-Length: <PayloadSizeBytes>
Date: 0
Content-Type: application/x-amz-json-1.0
```

Example

Using AWS query protocol

Sample Request

```
POST /177715257436/MyQueue/ HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Date: <Date>
Content-Type: application/x-www-form-urlencoded
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
Action=SetQueueAttributes
&Attribute.Name=Policy
&Attribute.Value=%7B%22Version%22%3A%222012-10-17%22%2C%22Id%22%3A%22UseCase1%22%2C
%22Statement%22%3A%5B%7B%22Sid%22%3A%22Queue1ReceiveMessage%22%2C%22Effect
%22%3A%22Allow%22%2C%22Principal%22%3A%7B%22AWS%22%3A%22*%22%7D%2C%22Action
%22%3A%22SQS%3AReceiveMessage%22%2C%22Resource%22%3A%22arn%3Aaws%3Asqs%3Aus-
east-1%3A555555555555%3AMyQueue6%22%7D%5D%7D
```

Sample Response

```
HTTP/1.1 200 OK
<?xml version="1.0"?>
<SetQueueAttributesResponse xmlns="http://queue.amazonaws.com/doc/2012-11-05/">
    <ResponseMetadata>
        <RequestId>5798727f-61f0-5457-99f0-2e0fa7fce329</RequestId>
    </ResponseMetadata>
</SetQueueAttributesResponse>
```

Example

The following example query request sets the visibility timeout to 35 seconds for a queue named MyQueue. The structure of AUTHPARAMS depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the *AWS General Reference*.

Note

An Amazon SQS message has three basic states:

1. Sent to a queue by a producer.
2. Received from the queue by a consumer.
3. Deleted from the queue.

A message is considered to be *stored* after it is sent to a queue by a producer, but not yet received from the queue by a consumer (that is, between states 1 and 2). There is no limit to the number of stored messages. A message is considered to be *in flight* after it is received from a queue by a consumer, but not yet deleted from the queue (that is, between states 2 and 3). There is a limit to the number of in flight messages.
Limits that apply to in flight messages are unrelated to the *unlimited* number of stored messages.
For most standard queues (depending on queue traffic and message backlog), there can be a maximum of approximately 120,000 in flight messages (received from a queue by a consumer, but not yet deleted from the queue). If you reach this limit, Amazon SQS returns the `OverLimit` error message. To avoid reaching the limit, you should delete messages from the queue after they're processed. You can also increase the number of queues you use to process your messages. To request a limit increase, file a support request.
For FIFO queues, there can be a maximum of 120,000 in flight messages (received from a queue by a consumer, but not yet deleted from the queue). If you reach this limit, Amazon SQS returns no error messages.

Using AWS JSON protocol (Default)

Sample Request

```
POST / HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Target: AmazonSQS.SetQueueAtrributes
X-Amz-Date: <Date>
Content-Type: application/x-amz-json-1.0
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive {
```

```
    "QueueUrl": "https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue/",
    "Attributes": {
        "VisibilityTimeout": "35"
    }
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <requestId>
Content-Length: <PayloadSizeBytes>
Date: 0
Content-Type: application/x-amz-json-1.0
```

Example

Using AWS query protocol

Sample Request

```
POST /177715257436/MyQueue/ HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Date: <Date>
Content-Type: application/x-www-form-urlencoded
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
Action=SetQueueAttributes
&Attribute.Name=VisibilityTimeout
&Attribute.Value=35
```

Sample Response

```
HTTP/1.1 200 OK
<?xml version="1.0"?>
<SetQueueAttributesResponse xmlns="http://queue.amazonaws.com/doc/2012-11-05/">
    <ResponseMetadata>
        <RequestId>e5cca473-4fc0-4198-a451-8abb94d02c75</RequestId>
    </ResponseMetadata>
</SetQueueAttributesResponse>
```

Example

The following example sets a queue named `MyDeadLetterQueue` as the dead-letter queue for a queue name `MySourceQueue` by calling the `SetQueueAttributes` action with the configuration details for the dead-letter queue.

> **Note**
>
> Queue URLs and names are case-sensitive.

Using AWS JSON protocol (Default)

Sample Request

```
POST / HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Target: AmazonSQS.SetQueueAtrributes
X-Amz-Date: <Date>
Content-Type: application/x-amz-json-1.0
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive {
    "QueueUrl": "https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue/",
    "Attributes": {
        "RedrivePolicy": "{\"maxReceiveCount\":\"5\",\"deadLetterTargetArn\":
\"arn:aws:sqs:us-east-1:555555555555:MyDeadLetterQueue\"}"
    }
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <requestId>
Content-Length: <PayloadSizeBytes>
Date: 0
Content-Type: application/x-amz-json-1.0
```

Example

Using AWS query protocol

Sample Request

```
POST /177715257436/MyQueue/ HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Date: <Date>
Content-Type: application/x-www-form-urlencoded
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
Action=SetQueueAttributes
&Attribute.Name=RedrivePolicy
&Attribute.Value=%7B%22maxReceiveCount%22%3A%225%22%2C%20%22deadLetterTargetArn%22%3A
%22arn%3Aaws%3Asqs%3Aus-east-1%3A555555555555%3AMyDeadLetterQueue%22%7D
```

Sample Response

```
HTTP/1.1 200 OK
<?xml version="1.0"?>
<SetQueueAttributesResponse xmlns="http://queue.amazonaws.com/doc/2012-11-05/">
    <ResponseMetadata>
        <RequestId>627e8ac6-73bf-515c-a359-d654eebaa6c3</RequestId>
    </ResponseMetadata>
</SetQueueAttributesResponse>
```

Example

The following example enables long polling by calling the `SetQueueAttributes` action with the `ReceiveMessageWaitTimeSeconds` parameter set to 20 seconds.

Using AWS JSON protocol (Default)

Sample Request

```
POST / HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Target: AmazonSQS.SetQueueAtrributes
X-Amz-Date: <Date>
Content-Type: application/x-amz-json-1.0
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive {
```

```
    "QueueUrl": "https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue/",
    "Attributes": {
        "ReceiveMessageWaitTimeSeconds": "20"
    }
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <requestId>
Content-Length: <PayloadSizeBytes>
Date: 0
Content-Type: application/x-amz-json-1.0
```

Example

Using AWS query protocol

Sample Request

```
POST /177715257436/MyQueue/ HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Date: <Date>
Content-Type: application/x-www-form-urlencoded
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
Action=SetQueueAttributes
&Attribute.Name=ReceiveMessageWaitTimeSeconds
&Attribute.Value=20
```

Sample Response

```
HTTP/1.1 200 OK
<?xml version="1.0"?>
<SetQueueAttributesResponse xmlns="http://queue.amazonaws.com/doc/2012-11-05/">
    <ResponseMetadata>
        <RequestId>3949c1a7-1e69-564c-ad00-9d3583174f09</RequestId>
    </ResponseMetadata>
</SetQueueAttributesResponse>
```

Example

The following example changes an existing queue into a delay queue by calling the `SetQueueAttributes` action with the `DelaySeconds` attribute set to 45 seconds. Changing the `DelaySeconds` attribute from its default value of `0` to a positive integer less than or equal to `900` changes the queue into a delay queue.

Using AWS JSON protocol (Default)

Sample Request

```
POST / HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Target: AmazonSQS.SetQueueAtrributes
X-Amz-Date: <Date>
Content-Type: application/x-amz-json-1.0
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive {
    "QueueUrl": "https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue/",
    "Attributes": {
        "DelaySeconds": "45"
    }
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <requestId>
Content-Length: <PayloadSizeBytes>
Date: 0
Content-Type: application/x-amz-json-1.0
```

Example

Using AWS query protocol

Sample Request

```
POST /177715257436/MyQueue/ HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
```

```
X-Amz-Date: <Date>
Content-Type: application/x-www-form-urlencoded
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
Action=SetQueueAttributes
&Attribute.Name=DelaySeconds
&Attribute.Value=45
```

Sample Response

```
HTTP/1.1 200 OK
<?xml version="1.0"?>
<SetQueueAttributesResponse xmlns="http://queue.amazonaws.com/doc/2012-11-05/">
    <ResponseMetadata>
        <RequestId>e4761152-39b6-556e-84a0-4dc0a78f4927</RequestId>
    </ResponseMetadata>
</SetQueueAttributesResponse>
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python (Boto3)
- AWS SDK for Ruby V3

StartMessageMoveTask

Starts an asynchronous task to move messages from a specified source queue to a specified destination queue.

Note

- This action is currently limited to supporting message redrive from queues that are configured as dead-letter queues (DLQs) of other Amazon SQS queues only. Non-SQS queue sources of dead-letter queues, such as AWS Lambda or Amazon SNS topics, are currently not supported.
- In dead-letter queues redrive context, the `StartMessageMoveTask` the source queue is the DLQ, while the destination queue can be the original source queue (from which the messages were driven to the dead-letter-queue), or a custom destination queue.
- Only one active message movement task is supported per queue at any given time.

Request Syntax

```
{
   "DestinationArn": "string",
   "MaxNumberOfMessagesPerSecond": number,
   "SourceArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

DestinationArn

The ARN of the queue that receives the moved messages. You can use this field to specify the destination queue where you would like to redrive messages. If this field is left blank, the messages will be redriven back to their respective original source queues.

Type: String

Required: No

MaxNumberOfMessagesPerSecond

The number of messages to be moved per second (the message movement rate). You can use this field to define a fixed message movement rate. The maximum value for messages per second is 500. If this field is left blank, the system will optimize the rate based on the queue message backlog size, which may vary throughout the duration of the message movement task.

Type: Integer

Required: No

SourceArn

The ARN of the queue that contains the messages to be moved to another queue. Currently, only ARNs of dead-letter queues (DLQs) whose sources are other Amazon SQS queues are accepted. DLQs whose sources are non-SQS queues, such as AWS Lambda or Amazon SNS topics, are not currently supported.

Type: String

Required: Yes

Response Syntax

```
{
   "TaskHandle": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

TaskHandle

An identifier associated with a message movement task. You can use this identifier to cancel a specified message movement task using the `CancelMessageMoveTask` action.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

InvalidAddress

The specified ID is invalid.

HTTP Status Code: 400

InvalidSecurity

The request was not made over HTTPS or did not use SigV4 for signing.

HTTP Status Code: 400

RequestThrottled

The request was denied due to request throttling.

- Exceeds the permitted request rate for the queue or for the recipient of the request.
- Ensure that the request rate is within the Amazon SQS limits for sending messages. For more information, see Amazon SQS quotas in the *Amazon SQS Developer Guide*.

HTTP Status Code: 400

ResourceNotFoundException

One or more specified resources don't exist.

HTTP Status Code: 400

UnsupportedOperation

Error code 400. Unsupported operation.

HTTP Status Code: 400

Examples

Example

Using AWS query protocol

The following example query starts a message move task on a dead-letter queue `MyDeadLetterQueue` to its source queue, with a limit of 10 messages per second. The structure of `AUTHPARAMS` depends on the signature of the API request. For more information, see Examples of Signed Signature Version 4 Requests in the *AWS General Reference*.

Sample Request

```
POST /  HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Date: <Date>
Content-Type: application/x-www-form-urlencoded
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
Action=StartMessageMoveTask
&SourceArn=arn:aws:sqs:us-east-1:555555555555:MyDeadLetterQueue
&MaxNumberOfMessagesPerSecond=10
```

Sample Response

```
HTTP/1.1 200 OK
<?xml version="1.0"?>
<StartMessageMoveTaskResponse xmlns="http://queue.amazonaws.com/doc/2012-11-05/">
    <StartMessageMoveTaskResult>

 <TaskHandle>eyJ0YXNrSWQiOiJkYzE2OWUwNC0wZTU1LTQ0ZDItYWE5MC1jMDgwY2ExZjM2ZjciLCJzb3VyY2VBcm4iOi
TaskHandle>
    </StartMessageMoveTaskResult>
    <ResponseMetadata>
        <RequestId>9b20926c-8b35-5d8e-9559-ce1c22e754dc</RequestId>
    </ResponseMetadata>
</StartMessageMoveTaskResponse>
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++

- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python (Boto3)
- AWS SDK for Ruby V3

TagQueue

Add cost allocation tags to the specified Amazon SQS queue. For an overview, see Tagging Your Amazon SQS Queues in the *Amazon SQS Developer Guide*.

When you use queue tags, keep the following guidelines in mind:

- Adding more than 50 tags to a queue isn't recommended.
- Tags don't have any semantic meaning. Amazon SQS interprets tags as character strings.
- Tags are case-sensitive.
- A new tag with a key identical to that of an existing tag overwrites the existing tag.

For a full list of tag restrictions, see Quotas related to queues in the *Amazon SQS Developer Guide*.

> **Note**
>
> Cross-account permissions don't apply to this action. For more information, see Grant cross-account permissions to a role and a username in the *Amazon SQS Developer Guide*.

Request Syntax

```
{
   "QueueUrl": "string",
   "Tags": {
      "string" : "string"
   }
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

QueueUrl

The URL of the queue.

Type: String

Required: Yes

Tags

The list of tags to be added to the specified queue.

Type: String to string map

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors.

InvalidAddress

The specified ID is invalid.

HTTP Status Code: 400

InvalidSecurity

The request was not made over HTTPS or did not use SigV4 for signing.

HTTP Status Code: 400

QueueDoesNotExist

Ensure that the `QueueUrl` is correct and that the queue has not been deleted.

HTTP Status Code: 400

RequestThrottled

The request was denied due to request throttling.

- Exceeds the permitted request rate for the queue or for the recipient of the request.

- Ensure that the request rate is within the Amazon SQS limits for sending messages. For more information, see Amazon SQS quotas in the *Amazon SQS Developer Guide*.

HTTP Status Code: 400

UnsupportedOperation

Error code 400. Unsupported operation.

HTTP Status Code: 400

Examples

This example illustrates one usage of TagQueue.

Example

Using AWS JSON protocol (Default)

Sample Request

```
POST / HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Target: AmazonSQS.TagQueue
X-Amz-Date: <Date>
Content-Type: application/x-amz-json-1.0
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
{
    "QueueUrl": "https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue/",
    "Tags": {
        "QueueType": "Production"
    }
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <requestId>
Content-Length: 0
Date: <Date>
```

```
Content-Type: application/x-amz-json-1.0
```

Example

Using AWS query protocol

Sample Request

```
POST /177715257436/MyQueue/ HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Date: <Date>
Content-Type: application/x-www-form-urlencoded
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
Action=TagQueue
&Tag.Key=QueueType
&Tag.Value=Production
```

Sample Response

```
HTTP/1.1 200 OK
<?xml version="1.0"?>
<TagQueueResponse xmlns="http://queue.amazonaws.com/doc/2012-11-05/">
    <ResponseMetadata>
        <RequestId>4bc96290-c3b5-5248-aace-3ee0056359b4</RequestId>
    </ResponseMetadata>
</TagQueueResponse>
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2

- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python (Boto3)
- AWS SDK for Ruby V3

UntagQueue

Remove cost allocation tags from the specified Amazon SQS queue. For an overview, see Tagging Your Amazon SQS Queues in the *Amazon SQS Developer Guide*.

> **Note**
>
> Cross-account permissions don't apply to this action. For more information, see Grant cross-account permissions to a role and a username in the *Amazon SQS Developer Guide*.

Request Syntax

```
{
   "QueueUrl": "string",
   "TagKeys": [ "string" ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

QueueUrl

The URL of the queue.

Type: String

Required: Yes

TagKeys

The list of tags to be removed from the specified queue.

Type: Array of strings

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors.

InvalidAddress

The specified ID is invalid.

HTTP Status Code: 400

InvalidSecurity

The request was not made over HTTPS or did not use SigV4 for signing.

HTTP Status Code: 400

QueueDoesNotExist

Ensure that the `QueueUrl` is correct and that the queue has not been deleted.

HTTP Status Code: 400

RequestThrottled

The request was denied due to request throttling.

- Exceeds the permitted request rate for the queue or for the recipient of the request.
- Ensure that the request rate is within the Amazon SQS limits for sending messages. For more information, see Amazon SQS quotas in the *Amazon SQS Developer Guide*.

HTTP Status Code: 400

UnsupportedOperation

Error code 400. Unsupported operation.

HTTP Status Code: 400

Examples

The following examples illustrate one usage of `UntagQueue`.

Example

Using AWS JSON protocol (Default)

Sample Request

```
POST / HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Target: AmazonSQS.UntagQueue
X-Amz-Date: <Date>
Content-Type: application/x-amz-json-1.0
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
{
    "QueueUrl": "https://sqs.us-east-1.amazonaws.com/177715257436/MyQueue/",
    "TagKeys": [
        "QueueType"
    ]
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <requestId>
Content-Length: 0
Date: <Date>
Content-Type: application/x-amz-json-1.0
```

Example

Using AWS query protocol

Sample Request

```
POST /177715257436/MyQueue/ HTTP/1.1
Host: sqs.us-east-1.amazonaws.com
X-Amz-Date: <Date>
Content-Type: application/x-www-form-urlencoded
Authorization: <AuthParams>
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
```

```
Action=UntagQueue
&TagKey=QueueType
```

Sample Response

```
HTTP/1.1 200 OK
<?xml version="1.0"?>
<UntagQueueResponse xmlns="http://queue.amazonaws.com/doc/2012-11-05/">
    <ResponseMetadata>
        <RequestId>2b4cc90a-f554-5f5b-a8ca-957a32378232</RequestId>
    </ResponseMetadata>
</UntagQueueResponse>
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python (Boto3)
- AWS SDK for Ruby V3

Data Types

The Amazon Simple Queue Service API contains several data types that various actions use. This section describes each data type in detail.

> **Note**
>
> The order of each element in a data type structure is not guaranteed. Applications should not assume a particular order.

The following data types are supported:

- BatchResultErrorEntry
- ChangeMessageVisibilityBatchRequestEntry
- ChangeMessageVisibilityBatchResultEntry
- DeleteMessageBatchRequestEntry
- DeleteMessageBatchResultEntry
- ListMessageMoveTasksResultEntry
- Message
- MessageAttributeValue
- MessageSystemAttributeValue
- SendMessageBatchRequestEntry
- SendMessageBatchResultEntry

BatchResultErrorEntry

Gives a detailed description of the result of an action on each entry in the request.

Contents

Code

An error code representing why the action failed on this entry.

Type: String

Required: Yes

Id

The `Id` of an entry in a batch request.

Type: String

Required: Yes

SenderFault

Specifies whether the error happened due to the caller of the batch API action.

Type: Boolean

Required: Yes

Message

A message explaining why the action failed on this entry.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++

- AWS SDK for Java V2
- AWS SDK for Ruby V3

ChangeMessageVisibilityBatchRequestEntry

Encloses a receipt handle and an entry ID for each message in `ChangeMessageVisibilityBatch`.

Contents

Id

An identifier for this particular receipt handle used to communicate the result.

> **Note**
>
> The `Id`s of a batch request need to be unique within a request.
> This identifier can have up to 80 characters. The following characters are accepted: alphanumeric characters, hyphens(-), and underscores (_).

Type: String

Required: Yes

ReceiptHandle

A receipt handle.

Type: String

Required: Yes

VisibilityTimeout

The new value (in seconds) for the message's visibility timeout.

Type: Integer

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ChangeMessageVisibilityBatchResultEntry

Encloses the `Id` of an entry in `ChangeMessageVisibilityBatch`.

Contents

Id

Represents a message whose visibility timeout has been changed successfully.

Type: String

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DeleteMessageBatchRequestEntry

Encloses a receipt handle and an identifier for it.

Contents

Id

The identifier for this particular receipt handle. This is used to communicate the result.

> **Note**
>
> The `Id`s of a batch request need to be unique within a request.
> This identifier can have up to 80 characters. The following characters are accepted: alphanumeric characters, hyphens(-), and underscores (_).

Type: String

Required: Yes

ReceiptHandle

A receipt handle.

Type: String

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DeleteMessageBatchResultEntry

Encloses the `Id` of an entry in `DeleteMessageBatch`.

Contents

Id

Represents a successfully deleted message.

Type: String

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ListMessageMoveTasksResultEntry

Contains the details of a message movement task.

Contents

ApproximateNumberOfMessagesMoved

The approximate number of messages already moved to the destination queue.

Type: Long

Required: No

ApproximateNumberOfMessagesToMove

The number of messages to be moved from the source queue. This number is obtained at the time of starting the message movement task and is only included after the message movement task is selected to start.

Type: Long

Required: No

DestinationArn

The ARN of the destination queue if it has been specified in the `StartMessageMoveTask` request. If a `DestinationArn` has not been specified in the `StartMessageMoveTask` request, this field value will be NULL.

Type: String

Required: No

FailureReason

The task failure reason (only included if the task status is FAILED).

Type: String

Required: No

MaxNumberOfMessagesPerSecond

The number of messages to be moved per second (the message movement rate), if it has been specified in the `StartMessageMoveTask` request. If a `MaxNumberOfMessagesPerSecond` has not been specified in the `StartMessageMoveTask` request, this field value will be NULL.

Type: Integer

Required: No

SourceArn

The ARN of the queue that contains the messages to be moved to another queue.

Type: String

Required: No

StartedTimestamp

The timestamp of starting the message movement task.

Type: Long

Required: No

Status

The status of the message movement task. Possible values are: RUNNING, COMPLETED, CANCELLING, CANCELLED, and FAILED.

Type: String

Required: No

TaskHandle

An identifier associated with a message movement task. When this field is returned in the response of the `ListMessageMoveTasks` action, it is only populated for tasks that are in RUNNING status.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Message

An Amazon SQS message.

Contents

Attributes

A map of the attributes requested in `ReceiveMessage` to their respective values. Supported attributes:

- `ApproximateReceiveCount`
- `ApproximateFirstReceiveTimestamp`
- `MessageDeduplicationId`
- `MessageGroupId`
- `SenderId`
- `SentTimestamp`
- `SequenceNumber`

`ApproximateFirstReceiveTimestamp` and `SentTimestamp` are each returned as an integer representing the epoch time in milliseconds.

Type: String to string map

Valid Keys: `All | SenderId | SentTimestamp | ApproximateReceiveCount | ApproximateFirstReceiveTimestamp | SequenceNumber | MessageDeduplicationId | MessageGroupId | AWSTraceHeader | DeadLetterQueueSourceArn`

Required: No

Body

The message's contents (not URL-encoded).

Type: String

Required: No

MD5OfBody

An MD5 digest of the non-URL-encoded message body string.

Type: String

Required: No

MD5OfMessageAttributes

An MD5 digest of the non-URL-encoded message attribute string. You can use this attribute to verify that Amazon SQS received the message correctly. Amazon SQS URL-decodes the message before creating the MD5 digest. For information about MD5, see RFC1321.

Type: String

Required: No

MessageAttributes

Each message attribute consists of a `Name`, `Type`, and `Value`. For more information, see Amazon SQS message attributes in the *Amazon SQS Developer Guide*.

Type: String to MessageAttributeValue object map

Required: No

MessageId

A unique identifier for the message. A `MessageId`is considered unique across all AWS accounts for an extended period of time.

Type: String

Required: No

ReceiptHandle

An identifier associated with the act of receiving the message. A new receipt handle is returned every time you receive a message. When deleting a message, you provide the last received receipt handle to delete the message.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

MessageAttributeValue

The user-specified message attribute value. For string data types, the `Value` attribute has the same restrictions on the content as the message body. For more information, see `SendMessage`.

`Name`, `type`, `value` and the message body must not be empty or null. All parts of the message attribute, including `Name`, `Type`, and `Value`, are part of the message size restriction (1 MiB or 1,048,576 bytes).

Contents

DataType

Amazon SQS supports the following logical data types: `String`, `Number`, and `Binary`. For the `Number` data type, you must use `StringValue`.

You can also append custom labels. For more information, see Amazon SQS Message Attributes in the *Amazon SQS Developer Guide*.

Type: String

Required: Yes

BinaryListValues

Not implemented. Reserved for future use.

Type: Array of Base64-encoded binary data objects

Required: No

BinaryValue

Binary type attributes can store any binary data, such as compressed data, encrypted data, or images.

Type: Base64-encoded binary data object

Required: No

StringListValues

Not implemented. Reserved for future use.

Type: Array of strings

Required: No

StringValue

Strings are Unicode with UTF-8 binary encoding. For a list of code values, see ASCII Printable Characters.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

MessageSystemAttributeValue

The user-specified message system attribute value. For string data types, the `Value` attribute has the same restrictions on the content as the message body. For more information, see `SendMessage`.

`Name`, `type`, `value` and the message body must not be empty or null.

Contents

DataType

Amazon SQS supports the following logical data types: `String`, `Number`, and `Binary`. For the `Number` data type, you must use `StringValue`.

You can also append custom labels. For more information, see Amazon SQS Message Attributes in the *Amazon SQS Developer Guide*.

Type: String

Required: Yes

BinaryListValues

Not implemented. Reserved for future use.

Type: Array of Base64-encoded binary data objects

Required: No

BinaryValue

Binary type attributes can store any binary data, such as compressed data, encrypted data, or images.

Type: Base64-encoded binary data object

Required: No

StringListValues

Not implemented. Reserved for future use.

Type: Array of strings

Required: No

StringValue

Strings are Unicode with UTF-8 binary encoding. For a list of code values, see ASCII Printable Characters.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

SendMessageBatchRequestEntry

Contains the details of a single Amazon SQS message along with an `Id`.

Contents

Id

An identifier for a message in this batch used to communicate the result.

> **Note**
>
> The `Id`s of a batch request need to be unique within a request.
> This identifier can have up to 80 characters. The following characters are accepted: alphanumeric characters, hyphens(-), and underscores (_).

Type: String

Required: Yes

MessageBody

The body of the message.

Type: String

Required: Yes

DelaySeconds

The length of time, in seconds, for which a specific message is delayed. Valid values: 0 to 900. Maximum: 15 minutes. Messages with a positive `DelaySeconds` value become available for processing after the delay period is finished. If you don't specify a value, the default value for the queue is applied.

> **Note**
>
> When you set `FifoQueue`, you can't set `DelaySeconds` per message. You can set this parameter only on a queue level.

Type: Integer

Required: No

MessageAttributes

Each message attribute consists of a `Name`, `Type`, and `Value`. For more information, see Amazon SQS message attributes in the *Amazon SQS Developer Guide*.

Type: String to MessageAttributeValue object map

Required: No

MessageDeduplicationId

This parameter applies only to FIFO (first-in-first-out) queues.

The token used for deduplication of messages within a 5-minute minimum deduplication interval. If a message with a particular `MessageDeduplicationId` is sent successfully, subsequent messages with the same `MessageDeduplicationId` are accepted successfully but aren't delivered. For more information, see Exactly-once processing in the *Amazon SQS Developer Guide*.

- Every message must have a unique `MessageDeduplicationId`,
  - You may provide a `MessageDeduplicationId` explicitly.
  - If you aren't able to provide a `MessageDeduplicationId` and you enable `ContentBasedDeduplication` for your queue, Amazon SQS uses a SHA-256 hash to generate the `MessageDeduplicationId` using the body of the message (but not the attributes of the message).
  - If you don't provide a `MessageDeduplicationId` and the queue doesn't have `ContentBasedDeduplication` set, the action fails with an error.
  - If the queue has `ContentBasedDeduplication` set, your `MessageDeduplicationId` overrides the generated one.
- When `ContentBasedDeduplication` is in effect, messages with identical content sent within the deduplication interval are treated as duplicates and only one copy of the message is delivered.
- If you send one message with `ContentBasedDeduplication` enabled and then another message with a `MessageDeduplicationId` that is the same as the one generated for the first `MessageDeduplicationId`, the two messages are treated as duplicates and only one copy of the message is delivered.

> **Note**
>
> The `MessageDeduplicationId` is available to the consumer of the message (this can be useful for troubleshooting delivery issues).
> If a message is sent successfully but the acknowledgement is lost and the message is resent with the same `MessageDeduplicationId` after the deduplication interval, Amazon SQS can't detect duplicate messages.
> Amazon SQS continues to keep track of the message deduplication ID even after the message is received and deleted.

The length of `MessageDeduplicationId` is 128 characters. `MessageDeduplicationId` can contain alphanumeric characters (`a-z`, `A-Z`, `0-9`) and punctuation (`!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~`).

For best practices of using `MessageDeduplicationId`, see Using the MessageDeduplicationId Property in the *Amazon SQS Developer Guide*.

Type: String

Required: No

MessageGroupId

`MessageGroupId` is an attribute used in Amazon SQS FIFO (First-In-First-Out) and standard queues. In FIFO queues, `MessageGroupId` organizes messages into distinct groups. Messages within the same message group are always processed one at a time, in strict order, ensuring that no two messages from the same group are processed simultaneously. In standard queues, using `MessageGroupId` enables fair queues. It is used to identify the tenant a message belongs to, helping maintain consistent message dwell time across all tenants during noisy neighbor events. Unlike FIFO queues, messages with the same `MessageGroupId` can be processed in parallel, maintaining the high throughput of standard queues.

- **FIFO queues:** `MessageGroupId` acts as the tag that specifies that a message belongs to a specific message group. Messages that belong to the same message group are processed in a FIFO manner (however, messages in different message groups might be processed out of order). To interleave multiple ordered streams within a single queue, use `MessageGroupId` values (for example, session data for multiple users). In this scenario, multiple consumers can process the queue, but the session data of each user is processed in a FIFO fashion.

If you do not provide a `MessageGroupId` when sending a message to a FIFO queue, the action fails.

`ReceiveMessage` might return messages with multiple `MessageGroupId` values. For each `MessageGroupId`, the messages are sorted by time sent.

- **Standard queues:**Use `MessageGroupId` in standard queues to enable fair queues. The `MessageGroupId` identifies the tenant a message belongs to. A tenant can be any entity that shares a queue with others, such as your customer, a client application, or a request type. When one tenant sends a disproportionately large volume of messages or has messages that require longer processing time, fair queues ensure other tenants' messages maintain low dwell time. This preserves quality of service for all tenants while maintaining the scalability and throughput of standard queues. We recommend that you include a `MessageGroupId` in all messages when using fair queues.

The length of `MessageGroupId` is 128 characters. Valid values: alphanumeric characters and punctuation `(!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~)`.

For best practices of using `MessageGroupId`, see Using the MessageGroupId Property in the *Amazon SQS Developer Guide*.

Type: String

Required: No

MessageSystemAttributes

The message system attribute to send Each message system attribute consists of a `Name`, `Type`, and `Value`.

> **Important**
>
> - Currently, the only supported message system attribute is `AWSTraceHeader`. Its type must be `String` and its value must be a correctly formatted AWS X-Ray trace header string.
> - The size of a message system attribute doesn't count towards the total size of a message.

Type: String to MessageSystemAttributeValue object map

Valid Keys: `AWSTraceHeader`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

SendMessageBatchResultEntry

Encloses a `MessageId` for a successfully-enqueued message in a `SendMessageBatch`.

Contents

Id

An identifier for the message in this batch.

Type: String

Required: Yes

MD5OfMessageBody

An MD5 digest of the non-URL-encoded message body string. You can use this attribute to verify that Amazon SQS received the message correctly. Amazon SQS URL-decodes the message before creating the MD5 digest. For information about MD5, see RFC1321.

Type: String

Required: Yes

MessageId

An identifier for the message.

Type: String

Required: Yes

MD5OfMessageAttributes

An MD5 digest of the non-URL-encoded message attribute string. You can use this attribute to verify that Amazon SQS received the message correctly. Amazon SQS URL-decodes the message before creating the MD5 digest. For information about MD5, see RFC1321.

Type: String

Required: No

MD5OfMessageSystemAttributes

An MD5 digest of the non-URL-encoded message system attribute string. You can use this attribute to verify that Amazon SQS received the message correctly. Amazon SQS URL-decodes the message before creating the MD5 digest. For information about MD5, see RFC1321.

Type: String

Required: No

SequenceNumber

This parameter applies only to FIFO (first-in-first-out) queues.

The large, non-consecutive number that Amazon SQS assigns to each message.

The length of `SequenceNumber` is 128 bits. As `SequenceNumber` continues to increase for a particular `MessageGroupId`.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Common Parameters

The following list contains the parameters that all actions use for signing Signature Version 4 requests with a query string. Any action-specific parameters are listed in the topic for that action. For more information about Signature Version 4, see Signing AWS API requests in the *IAM User Guide*.

X-Amz-Algorithm

The hash algorithm that you used to create the request signature.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Valid Values: `AWS4-HMAC-SHA256`

Required: Conditional

X-Amz-Credential

The credential scope value, which is a string that includes your access key, the date, the region you are targeting, the service you are requesting, and a termination string ("aws4_request"). The value is expressed in the following format: *access_key*/*YYYYMMDD*/*region*/*service*/aws4_request.

For more information, see Create a signed AWS API request in the *IAM User Guide*.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

X-Amz-Date

The date that is used to create the signature. The format must be ISO 8601 basic format (YYYYMMDD'T'HHMMSS'Z'). For example, the following date time is a valid X-Amz-Date value: `20120325T120000Z`.

Condition: X-Amz-Date is optional for all requests; it can be used to override the date used for signing requests. If the Date header is specified in the ISO 8601 basic format, X-Amz-Date is not required. When X-Amz-Date is used, it always overrides the value of the Date header. For more information, see Elements of an AWS API request signature in the *IAM User Guide*.

Type: string

Required: Conditional

X-Amz-Security-Token

The temporary security token that was obtained through a call to AWS Security Token Service (AWS STS). For a list of services that support temporary security credentials from AWS STS, see AWS services that work with IAM in the *IAM User Guide*.

Condition: If you're using temporary security credentials from AWS STS, you must include the security token.

Type: string

Required: Conditional

X-Amz-Signature

Specifies the hex-encoded signature that was calculated from the string to sign and the derived signing key.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

X-Amz-SignedHeaders

Specifies all the HTTP headers that were included as part of the canonical request. For more information about specifying signed headers, see Create a signed AWS API request in the *IAM User Guide*.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

Common Errors

This section lists the errors common to the API actions of all AWS services. For errors specific to an API action for this service, see the topic for that API action.

AccessDeniedException

You do not have sufficient access to perform this action.

HTTP Status Code: 400

IncompleteSignature

The request signature does not conform to AWS standards.

HTTP Status Code: 400

InternalFailure

The request processing has failed because of an unknown error, exception or failure.

HTTP Status Code: 500

InvalidAction

The action or operation requested is invalid. Verify that the action is typed correctly.

HTTP Status Code: 400

InvalidClientTokenId

The X.509 certificate or AWS access key ID provided does not exist in our records.

HTTP Status Code: 403

InvalidParameterCombination

Parameters that must not be used together were used together.

HTTP Status Code: 400

InvalidParameterValue

An invalid or out-of-range value was supplied for the input parameter.

HTTP Status Code: 400

InvalidQueryParameter

The AWS query string is malformed or does not adhere to AWS standards.

HTTP Status Code: 400

MalformedQueryString

The query string contains a syntax error.

HTTP Status Code: 404

MissingAction

The request is missing an action or a required parameter.

HTTP Status Code: 400

MissingAuthenticationToken

The request must contain either a valid (registered) AWS access key ID or X.509 certificate.

HTTP Status Code: 403

MissingParameter

A required parameter for the specified action is not supplied.

HTTP Status Code: 400

NotAuthorized

You do not have permission to perform this action.

HTTP Status Code: 400

OptInRequired

The AWS access key ID needs a subscription for the service.

HTTP Status Code: 403

RequestExpired

The request reached the service more than 15 minutes after the date stamp on the request or more than 15 minutes after the request expiration date (such as for pre-signed URLs), or the date stamp on the request is more than 15 minutes in the future.

HTTP Status Code: 400

ServiceUnavailable

The request has failed due to a temporary failure of the server.

HTTP Status Code: 503

ThrottlingException

The request was denied due to request throttling.

HTTP Status Code: 403

ValidationError

The input fails to satisfy the constraints specified by an AWS service.

HTTP Status Code: 400