Amazon Simple Queue Service

API Reference

API Version 2011-10-01

Amazon Web Services

Amazon Simple Queue Service: API Reference

Amazon Web Services

Welcome

This is the *Amazon Simple Queue Service* API Reference. This section describes who should read this guide, how the guide is organized, and other resources related to the Amazon Simple Queue Service (Amazon SQS).

Amazon SQS offers reliable and scalable hosted queues for storing messages as they travel between computers. By using Amazon SQS, you can move data between distributed components of your applications that perform different tasks without losing messages or requiring each component to be always available.

Current WSDL (2011-10-01)	Location of the current WSDL.
Making API Requests	Information about using the API.
Actions (p. 5)	List of the Amazon SQS actions by function.
Common Query Parameters (p. 53)	Parameters that all Query actions can use
Common Errors (p. 54)	Client and server errors the API returns
Amazon SQS product page	Information about this product
Regions and Endpoints	Regions and endpoints for Amazon SQS

API Versions

The version of an API is specified as a date, such as 2011-10-01. The following table displays the actions associated with each API version.

Action	2008-01-01	2009-02-01	2011-10-01
`CreateQueue`	✓	✓	✓
`ListQueues`	✓	✓	✓

Action	2008-01-01	2009-02-01	2011-10-01
DeleteQueue	✓	✓	✓
GetQueueAttributes	✓	✓	✓
SetQueueAttributes	✓	✓	✓
SendMessage	✓	✓	✓
ReceiveMessage	✓	✓	✓
DeleteMessage	✓	✓	✓
AddPermission		✓	✓
RemovePermission		✓	✓
ChangeMessageVisibility		✓	✓
GetQueueUrl			✓
SendMessageBatch			✓
DeleteMessageBatch			✓
ChangeMessageVisibilityBatch			✓

The version of the API can be found in the URLs of the WSDL file. It can also be found in the target namespace of the WSDL file.

You can retrieve the WSDL for a previous version of the API by replacing the version date in the URL with the desired version. For example, the following retrieves the WSDL for API version 2008-01-01.

http://sqs.us-east-1.amazonaws.com/doc/2008-01-01/QueueService.wsdl

Note to Users of the 2008-01-01 API Version

With the release of version 2009-02-01 we changed how you reference a queue. In the 2008-01-01 API version you used the following structure:

```
http://sqs.us-east-1.amazonaws.com/<queuename>
```

In the 2009-02-01 and 2011-10-01 API versions, you must insert the queue owner's AWS account number before the queue name in all service requests other than CreateQueue (p. 15) and ListQueues (p. 34):

```
http://sqs.us-east-1.amazonaws.com/<acct#>/<queuename>
```

Specifying the API Version

For all requests, you must specify the API version you want to use with the *Version* parameter. Specifying the version ensures that the service does not return response elements that your application is not designed to handle.

The following example specifies the latest API version with the *Version* parameter.

```
http://sqs.us-east-1.amazonaws.com/
?Action=CreateQueue
&QueueName=queue2
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&Version=2011-10-01
&Expires=2011-10-10T12:00:00Z
&Signature=Dqlp3Sd6ljTUA9Uf6SGtEExwUQEXAMPLE
&SignatureVersion=2
&SignatureMethod=HmacSHA256
```

List of Actions by Function

Actions for Queues

- CreateQueue (p. 15)
- DeleteQueue (p. 24)
- ListQueues (p. 34)
- GetQueueUrl (p. 32)
- GetQueueAttributes (p. 26)
- SetQueueAttributes (p. 49)

Actions for Access Control on Queues

- AddPermission (p. 6)
- RemovePermission (p. 40)

Actions for Messages

- SendMessage (p. 42)
- SendMessageBatch (p. 45)
- ReceiveMessage (p. 36)
- DeleteMessage (p. 19)
- DeleteMessageBatch (p. 21)
- ChangeMessageVisibility (p. 9)
- ChangeMessageVisibilityBatch (p. 11)

Actions

Topics

Note

AWS provides libraries, sample code, tutorials, and other resources for software developers who prefer to build applications using language-specific APIs instead of Amazon SQS's Query API. These libraries provide basic functions (not included in the Query API), such as request authentication, request retries, and error handling so you can get started more easily. Libraries and resources are available for the following languages:

- Java
- PHP
- Ruby
- Windows and .NET

For libraries and sample code in all languages, go to Sample Code & Libraries.

AddPermission

Description

The `AddPermission` action adds a permission to a queue for a specific principal. This allows for sharing access to the queue.

When you create a queue, you have full control access rights for the queue. Only you (as owner of the queue) can grant or deny permissions to the queue. For more information about these permissions, see Shared Queues in the Amazon SQS Developer Guide.

Note

`AddPermission` writes an SQS-generated policy. If you want to write your own policy, use SetQueueAttributes (p. 49) to upload your policy. For more information about writing your own policy, see Appendix: The Access Policy Language in the Amazon SQS Developer Guide.

Request Parameters

The following table lists the special request parameters the `AddPermission` action uses in addition to the common parameters all actions use (for more information, see Common Query Parameters (p. 53)).

Note

Some API operations take lists of parameters. These lists are specified using the `param.n` notation. Values of n are integers starting from 1. For example, a parameter list with two elements looks like this:

```
&Attribute.1=this
&Attribute.2=that
```

Name	Description	Required
Label	The unique identification of the permission you're setting. Type: String (example: `AliceSendMessage`) Constraints: Maximum 80 characters; alphanumeric characters, hyphens (-), and underscores (_) are allowed. Default: None	Yes

Name	Description	Required
AWSAccountId.n	The AWS account number of the principal who will be given permission. The principal must have an AWS account, but does not need to be signed up for Amazon SQS. For information about locating the AWS account identification, see Your AWS Identifiers in the Amazon SQS Developer Guide. Type: String Constraints: Valid 12-digit AWS account number, without hyphens Default: None	Yes
ActionName.n	The action you want to allow for the specified principal. For more information about these actions, see Understanding Permissions in the Amazon SQS Developer Guide. Type: String Valid values: `*` | `SendMessage` | `ReceiveMessage` | `DeleteMessage` | `ChangeMessageVisibility` | `GetQueueAttributes` | `GetQueueUrl` Default: None	Yes

Note

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

Response Elements

The `AddPermission` response only returns metadata. For more information, see Responses in the Amazon SQS Developer Guide.

Examples

The following example Query request grants a *`SendMessage`* permission to the principal whose AWS account number is 125074342641.

Sample Request

```
http://sqs.us-east-1.amazonaws.com/123456789012/testQueue/
?Action=AddPermission
&Label=testLabel
&AWSAccountId.1=125074342641
&ActionName.1=SendMessage
&AWSAccountId.2=125074342642
&ActionName.2=ReceiveMessage
&Version=2009-02-01
&SignatureMethod=HmacSHA256
&Expires=2009-04-18T22%3A52%3A43PST
```

```
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&SignatureVersion=2
&Signature=Dqlp3Sd6ljTUA9Uf6SGtEExwUQEXAMPLE
```

Sample Response

```
<AddPermissionResponse>
    <ResponseMetadata>
        <RequestId>
            9a285199-c8d6-47c2-bdb2-314cb47d599d
        </RequestId>
    </ResponseMetadata>
</AddPermissionResponse>
```

Related Actions

- RemovePermission (p. 40)

ChangeMessageVisibility

Description

The `ChangeMessageVisibility` action changes the visibility timeout of a specified message in a queue to a new value. The maximum allowed timeout value you can set the value to is 12 hours. This means you can't extend the timeout of a message in an existing queue to more than a total visibility timeout of 12 hours. (For more information visibility timeout, see Visibility Timeout in the Amazon SQS Developer Guide.)

For example, let's say the timeout for the queue is 30 seconds, and you receive a message. Once you're 20 seconds into the timeout for that message (i.e., you have 10 seconds left), you extend it by 60 seconds by calling `ChangeMessageVisibility` with *`VisibilityTimeoutset`* to 60 seconds. You have then changed the remaining visibility timeout from 10 seconds to 60 seconds.

Important

If you attempt to set the *`VisibilityTimeout`* to an amount more than the maximum time left, Amazon SQS returns an error. It will not automatically recalculate and increase the timeout to the maximum time remaining.

Important

Unlike with a queue, when you change the visibility timeout for a specific message, that timeout value is applied immediately but is not saved in memory for that message. If you don't delete a message after it is received, the visibility timeout for the message the next time it is received reverts to the original timeout value, not the value you set with the `ChangeMessageVisibility` action.

Request Parameters

The following table lists the special request parameters the action uses (in addition to the common request parameters listed in Common Query Parameters (p. 53)).

Name	Description	Required
`ReceiptHandle`	The receipt handle associated with the message whose visibility timeout you want to change. This parameter is returned by the ReceiveMessage (p. 36) action. Type: String Length Constraint: Maximum 1024 characters Default: None	Yes
`VisibilityTimeout`	The new value for the message's visibility timeout (in seconds). Type: Integer from 0 to 43200 (maximum 12 hours) Constraints: This value is limited to 43200 seconds (12 hours) Default: None	Yes

Response Elements

The response contains no special elements besides the common elements listed in the Responses section in the Amazon SQS Developer Guide.

Special Errors

The action returns no errors besides the common errors listed in Common Errors (p. 54).

Examples

The following example Query request changes the visibility timeout for a message to 60 seconds.

Sample Request

```
http://sqs.us-east-1.amazonaws.com/123456789012/testQueue/
?Action=ChangeMessageVisibility
&VisibilityTimeout=60
&ReceiptHandle=MbZj6wDWli%2BJvwwJaBV%2B3dcjk2YW2vA3%2BSTFFljT
M8tJJg6HRG6PYSasuWXPJB%2BCwLj1FjgXUv1uSj1gUPAWV66FU/WeR4mq2OKpEGY
WbnLmpRCJVAyeMjeU5ZBdtcQ%2BQEauMZc8ZRv37sIW2iJKq3M9MFx1YvV11A2x/K
SbkJ0=
&Version=2009-02-01
&SignatureMethod=HmacSHA256
&Expires=2009-04-18T22%3A52%3A43PST
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&SignatureVersion=2
&Signature=Dqlp3Sd6ljTUA9Uf6SGtEExwUQEXAMPLE
```

Sample Response

```
<ChangeMessageVisibilityResponse>
    <ResponseMetadata>
        <RequestId>
            6a7a282a-d013-4a59-aba9-335b0fa48bed
        </RequestId>
    </ResponseMetadata>
</ChangeMessageVisibilityResponse>
```

Related Actions

- DeleteMessage (p. 19)
- ReceiveMessage (p. 36)

ChangeMessageVisibilityBatch

Description

The `ChangeMessageVisibilityBatch` action is a batch version of the ChangeMessageVisibility (p. 9) action. 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.

Note

Some API operations take lists of parameters. These lists are specified using the `param.n` notation. Values of n are integers starting from 1. For example, a parameter list with two elements looks like this:

```
&Attribute.1=this
&Attribute.2=that
```

Request Parameters

The following table lists the special request parameters that the `ChangeMessageVisibilityBatch` action uses in addition to the common request parameters that all actions use (for more information, see Common Query Parameters (p. 53)).

Important

All of the following parameters are list parameters that must be prefixed with `ChangeMessageVisibilityBatchRequestEntry.n`, where n is an integer value starting with 1. For example, a parameter list for this action might look like this:

```
&ChangeMessageVisibilityBatchRequestEntry.1.Id=change_visibility_msg_2
&ChangeMessageVisibilityBatchRequestEntry.1.ReceiptHandle=Your_Re
ceipt_Handle
&ChangeMessageVisibilityBatchRequestEntry.1.VisibilityTimeout=45
```

Name	Description	Required
Id	An identifier that you assign to the request. Requests that are part of the same call to `ChangeMessageVisibilityBatch` cannot have the same identifier. Type: String.	Yes

Name	Description	Required
ReceiptHandle	The receipt handle associated with the message that has the visibility timeout that you want to change. This parameter is returned by the ReceiveMessage action. Type: String. Length Constraint: Maximum 1024 characters	Yes
VisibilityTimeout	The new value for the message's visibility timeout (in seconds). Type: Integer from 0 to 43200 (maximum 12 hours). Default: If a value is not provided, no change is made to this attribute.	No

Response Elements

For each message in the batch, the response contains a *ChangeMessageVisibilityBatchResultEntry* tag if the message succeeds or a *BatchResultErrorEntry* tag if the message fails.

The following table lists the response elements that the ChangeMessageVisibilityBatch action returns in addition to the common response elements that all actions return (for more information, see Responses section in the Amazon SQS Developer Guide.

Name	Description
ChangeMessageVisibilityBatchResultEntry	An element containing a successfully modified message's information. Ancestor: ChangeMessageVisibilityBatchResult Children: • **Id—**The Id name that you assigned to the message.
BatchResultErrorEntry	An element containing information about a message that did not have its message visibility value modified. Ancestor: ChangeMessageVisibilityBatchResult Children: • **Id—**The Id name that you assigned to the message. • **SenderFault—**A boolean value. • **Code—**A short string description of the error. • **Message—**A description of the error.

Special Errors

Because batch requests can result in a combination of successful and unsuccessful actions, an HTTP status code of 200 might include a `BatchResultErrorEntry`.

The following table lists the special errors that the `ChangeMessageVisibilityBatch` action returns in addition to the common errors that all actions return (for more information, see Common Errors (p. 54)).

Error	Description	HTTP Status Code
`BatchResultErrorEntry`	The visibility timeout of a message was not changed. For more information, see the section called “Response Elements” (p. 12).	200

Examples

Sample Request

The following example `ChangeMessageVisibilityBatch` request changes the visibility timeout settings for two messages. You must URL encode the entire URL; however, we've URL encoded only the message body to make the example easier for you to read.

```
http://sqs.us-east-1.amazonaws.com/123456789012/testQueue/
&Action=ChangeMessageVisibilityBatch
&Version=2011-10-01
&ChangeMessageVisibilityBatchRequestEntry.1.Id=change_visibility_msg_2
&ChangeMessageVisibilityBatchRequestEntry.1.Re
ceiptHandle=gfk0T0R0waama4fVFffkjKzmhMCymjQvfTFk2LxT33G4ms5subrE0deLK
WSscPU1oD3J9zgeS4PQQ3U30qOumIE6AdAv3w%2F%2Fa1IXW6AqaWhGsEPaLm3Vf6Ii
WqdM8u5imB%2BNTwj3tQRzOWdTOePjOjPcTpRxBtXix%2BEvwJOZUma9wabv%2BSw6ZHjwm
NcVDx8dZXJhVp16Bksiox%2FGrUvrVTCJRTWTLc59oHLLF8sEkKzRmGNzTDGTiV%2BYjH
fQj60FD3rVaXmzTsoNxRhKJ72uIHVMGVQiAGgBX6HGv9LDmYhPXw4hy%2FNgIg%3D%3D
&ChangeMessageVisibilityBatchRequestEntry.1.VisibilityTimeout=45
&ChangeMessageVisibilityBatchRequestEntry.2.Id=change_visibility_msg_3
&ChangeMessageVisibilityBatchRequestEntry.2.Re
ceiptHandle=gfk0T0R0waama4fVFffkjKzmhMCymjQvfTFk2LxT33FUgBz3%2BnougdeLK
WSscPU1%2FXgx%2BxcNnjnQQ3U30qOumIE6AdAv3w%2F%2Fa1IXW6AqaWhGsEPaLm3Vf6Ii
WqdM8u5imB%2BNTwj3tQRzOWdTOePjOsogjZM%2F7kzn4Ew27XLU9I%2FYaWYmKvDbq%2Fk3HK
VB9HfB43kE49atP2aWrzNL4yunG41Q4cfRRtfJdcGQGNHQ2%2Byd0Usf5qR1dZr1iDo5xk946eQat83Ax
TRP%2BY4Qi0V7FAeSLH9su9xpX6HGv9LDmYhPXw4hy%2FNgIg%3D%3D
&ChangeMessageVisibilityBatchRequestEntry.2.VisibilityTimeout=45
&SignatureMethod=HmacSHA256
&Expires=2011-10-18T22%3A52%3A43PST
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&SignatureVersion=2
&Signature=Dqlp3Sd6ljTUA9Uf6SGtEExwUQEXAMPLE
```

Sample Response

```
<ChangeMessageVisibilityBatchResponse>
    <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>
```

Related Actions

- ChangeMessageVisibility (p. 9)

CreateQueue

Description

The `CreateQueue` action creates a new queue.

When you request `CreateQueue`, you provide a name for the queue. To successfully create a *new* queue, you must provide a name that is unique within the scope of your own queues.

Note

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

If you provide the name of an existing queue, along with the exact names and values of all the queue's attributes, `CreateQueue` returns the queue URL for the existing queue. If the queue name, attribute names, or attribute values do not match an existing queue, `CreateQueue` returns an error.

Tip

Use GetQueueUrl (p. 32) to get a queue's URL. `GetQueueUrl` requires only the *`QueueName`* parameter.

The default value for the queue's attributes are set when the queue is created. You can override these values with the *`Attribute.n.name`* and *`Attribute.n.value`* request parameters. For the default values of each attribute, see the table of parameters in Request Parameters (p. 15).

Note

Some API operations take lists of parameters. These lists are specified using the `param.n` notation. Values of n are integers starting from 1. For example, a parameter list with two elements looks like this:

```
&Attribute.1=this
&Attribute.2=that
```

Request Parameters

The following table lists the special request parameters the `CreateQueue` action uses in addition to the common parameters all actions use (for more information, see Common Query Parameters (p. 53)).

Name	Description	Required
`QueueName`	The name to use for the queue created. Type: String Constraints: Maximum 80 characters; alphanumeric characters, hyphens (-), and underscores (_) are allowed.	Yes

Name	Description	Required
Attribute.n.Name	The name of the attribute you want to set. • **VisibilityTimeout—**The length of time (in seconds) that a message received from a queue will be invisible to other receiving components when they ask to receive messages. For more information about `VisibilityTimeout`, see Visibility Timeout in the Amazon SQS Developer Guide. • **Policy—**The formal description of the permissions for a resource. For more information about `Policy`, see Basic Policy Structure in the Amazon SQS Developer Guide. • **MaximumMessageSize—**The limit of how many bytes a message can contain before Amazon SQS rejects it. • **MessageRetentionPeriod—**The number of seconds Amazon SQS retains a message. • **DelaySeconds—**The time in seconds that the delivery of all messages in the queue will be delayed.	No
Attribute.n.Value	The value of the attribute you want to set. Constraints: Constraints are specific for each value. • **VisibilityTimeout—**An integer from 0 to 43200 (12 hours). The default for this attribute is 30. • **Policy—**A valid form-url-encoded policy. For more information about policy structure, see Basic Policy Structure in the Amazon SQS Developer Guide. For more information about form-url-encoding, see http://www.w3.org/MarkUp/html-spec/html-spec_8.html#SEC8.2.1. • **MaximumMessageSize—**An integer from 1024 bytes (1 KiB) up to 65536 bytes (64 KiB). The default for this attribute is 65536 (64 KiB). • **MessageRetentionPeriod—**Integer representing seconds, from 60 (1 minute) to 1209600 (14 days). The default for this attribute is 345600 (4 days). • **DelaySeconds—**An integer from 0 to 900 (15 minutes). The default for this attribute is 0 (zero). Default: Varies according to attribute	Yes, if there is a corresponding Name Attribute.n.name parameter

Response Elements

The following table lists the elements that the `CreateQueue` response includes in addition to the elements that are returned in all successful responses (for more information, see Responses section in the Amazon SQS Developer Guide.

Name	Description
QueueUrl	The queue URL for the queue that you created. Type: String Ancestor: CreateQueueResult

Special Errors

The following table lists the special errors that the CreateQueue action returns in addition to the common errors that all actions return (for more information, see Common Errors (p. 54)).

Error	Description	HTTP Status Code
AWS.SimpleQueueService.QueueDeletedRecently	You must wait 60 seconds after deleting a queue before you can create another with the same name.	400
AWS.SimpleQueueService.QueueNameExists	Queue already exists. SQS returns this error only if the request includes an attribute name or value that differs from the name or value for the existing attribute.	400

Examples

The following example Query request creates a new queue named queue2.

Sample Request

```
http://sqs.us-east-1.amazonaws.com/
?Action=CreateQueue
&QueueName=testQueue
&Attribute.1.Name=VisibilityTimeout
&Attribute.1.Value=40
&Version=2011-10-01
&SignatureMethod=HmacSHA256
&Expires=2011-10-18T22%3A52%3A43PST
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&SignatureVersion=2
&Signature=Dqlp3Sd6ljTUA9Uf6SGtEExwUQEXAMPLE
```

Sample Response

```
<CreateQueueResponse>
    <CreateQueueResult>
        <QueueUrl>
```

```
            http://sqs.us-east-1.amazonaws.com/123456789012/testQueue
        </QueueUrl>
    </CreateQueueResult>
    <ResponseMetadata>
        <RequestId>
            7a62c49f-347e-4fc4-9331-6e8e7a96aa73
        </RequestId>
    </ResponseMetadata>
</CreateQueueResponse>
```

Related Actions

- DeleteQueue (p. 24)
- ListQueues (p. 34)
- SetQueueAttributes (p. 49)

DeleteMessage

Description

The `DeleteMessage` action deletes the specified message from the specified queue. You specify the message by using the message's receipt handle and not the message ID you received when you sent the message. Even if the message is locked by another reader due to the visibility timeout setting, it is still deleted from the queue. If you leave a message in the queue for more than 4 days, SQS automatically deletes it.

Note

The receipt handle is associated with a specific instance of receiving the message. If you receive a message more than once, the receipt handle you get each time you receive the message is different. When you request `DeleteMessage`, if you don't provide the most recently received receipt handle for the message, the request will still succeed, but the message might not be deleted.

Important

It is possible you will receive a message even after you have deleted it. This might happen on rare occasions if one of the servers storing a copy of the message is unavailable when you request to delete the message. The copy remains on the server and might be returned to you again on a subsequent receive request. You should create your system to be idempotent so that receiving a particular message more than once is not a problem.

Request Parameters

The following table lists the special request parameters the `DeleteMessage` action uses in addition to the common parameters all actions use (for more information, see Common Query Parameters (p. 53)).

Name	Description	Required
`ReceiptHandle`	The receipt handle associated with the message you want to delete. Type: String Length Constraint: Maximum 1024 characters	Yes

Response Elements

The response contains no special elements besides the common elements returned in all successful responses (for more information, see Responses section in the Amazon SQS Developer Guide.

Special Errors

The request doesn't fail unless the *`ReceiptHandle`* is malformed. Even if the specified *`ReceiptHandle`* doesn't exist or isn't the most recently returned receipt handle for that message, the action returns `Success`.

Examples

The following example Query request deletes a message from the specified queue.

Sample Request

```
http://sqs.us-east-1.amazonaws.com/123456789012/testQueue/
?Action=DeleteMessage
&ReceiptHandle=MbZj6wDWli%2BJvwwJaBV%2B3dcjk2YW2vA3%2BSTFFljT
M8tJJg6HRG6PYSasuWXPJB%2BCwLj1FjgXUv1uSj1gUPAWV66FU/WeR4mq2OKpEGY
WbnLmpRCJVAyeMjeU5ZBdtcQ%2BQEauMZc8ZRv37sIW2iJKq3M9MFx1YvV11A2x/K
SbkJ0=
&Version=2009-02-01
&SignatureMethod=HmacSHA256
&Expires=2009-04-18T22%3A52%3A43PST
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&SignatureVersion=2
&Signature=Dqlp3Sd6ljTUA9Uf6SGtEExwUQEXAMPLE
```

Sample Response

```
<DeleteMessageResponse>
    <ResponseMetadata>
        <RequestId>
            b5293cb5-d306-4a17-9048-b263635abe42
        </RequestId>
    </ResponseMetadata>
</DeleteMessageResponse>
```

Related Actions

- ReceiveMessage (p. 36)
- SendMessage (p. 42)

DeleteMessageBatch

Description

The `DeleteMessageBatch` action is a batch version of the DeleteMessage (p. 19) 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.

Note

Some API operations take lists of parameters. These lists are specified using the `param.n` notation. Values of n are integers starting from 1. For example, a parameter list with two elements looks like this:

```
&Attribute.1=this
&Attribute.2=that
```

Request Parameters

The following table lists the special request parameters that the `DeleteMessageBatch` action uses in addition to the common request parameters that all actions use (for more information, see Common Query Parameters (p. 53)).

Name	Description	Required
`DeleteMessageBatchRequestEntry.n.Id`	An identifier that you assign to the message. Requests that are part of the same call to `DeleteMessageBatch` cannot have the same identifier. Type: String.	Yes
`DeleteMessageBatchRequestEntry.n.ReceiptHandle`	The receipt handle that is associated with the message that you want to delete. This parameter is returned by the ReceiveMessage action. Type: String. Length Constraint: Maximum 1024 characters	Yes

Response Elements

For each message in the batch, the response contains a *`DeleteMessageBatchResultEntry`* tag if the message is deleted or a *`BatchResultErrorEntry`* tag if the message cannot be deleted.

The following table lists the response elements that the `DeleteMessageBatch` action returns in addition to the common response elements that all actions return (for more information, see the Responses section in the Amazon SQS Developer Guide.

Name	Description
`DeleteMessageBatchResultEntry`	An element containing a successfully deleted message's information. Ancestor: `DeleteMessageBatchResult` Children: • **Id—**The Id name that you assigned to the message.
`BatchResultErrorEntry`	An element containing information about a message that could not be deleted. Ancestor: `DeleteMessageBatchResult` Children: • **Id—**The Id name that you assigned to the message. • **SenderFault—**A boolean value. • **Code—**A short string description of the error. • **Message—**A description of the error.

Special Errors

The request doesn't fail unless the *`ReceiptHandle`* is malformed. Even if the specified *`ReceiptHandle`* doesn't exist or isn't the most recently returned receipt handle for that message, the action returns `Success`.

Examples

Sample Request

The following example `DeleteMessageBatch` request deletes two messages. You must URL encode the entire URL; however, we've URL encoded only the message body to make the example easier for you to read.

```
http://sqs.us-east-1.amazonaws.com/123456789012/testQueue/
&Action=DeleteMessageBatch
&Version=2011-10-01
&DeleteMessageBatchRequestEntry.1.Id=msg1
&DeleteMessageBatchRequestEntry.1.ReceiptHandle=gfk0T0R0waama4fVFffkjPQrrvzM
rOg0fTFk2LxT33EuB8wR0ZCFgKWyXGWFoqqpCIiprQUEhir%2F5LeGPpYTLzjqLQxyQYaQALeS
NHb0us3uE84uujxpBhsDkZUQkjFFkNqBXn48xlMcVhTcI3YLH%2Bd%2BIqetIOHgBCZAPx6r%2B09dW
aBXei6nbK5Ygih21DCDdAwFV68Jo8DXhb3ErEfoDqx7vyvC5nCp
dwqv%2BJhU%2FTNGjNN8t51v5c%2FAXvQsAzyZVNapxUrHIt4NxRhKJ72uICcxruyE8eRXlxIVN
geNP8ZEDcw7zZU1Zw%3D%3D
&DeleteMessageBatchRequestEntry.2.Id=msg2
&DeleteMessageBatchRequestEntry.2.ReceiptHandle=gfk0T0R0waama4fVFffkjKzmhM
CymjQvfTFk2LxT33G4ms5subrE0deLKWSscPU1oD3J9zgeS4PQQ3U30qOu
mIE6AdAv3w%2F%2Fa1IXW6AqaWhGsEPaLm3Vf6IiWqdM8u5imB%2BNTwj3tQRzOWdTOePjOjPcTpRxB
```

```
tXix%2BEvwJOZUma9wabv%2BSw6ZHjwmNcVDx8dZXJhVp16Bksi
ox%2FGrUvrVTCJRTWTLc59oHLLF8sEkKzRmGNzTDGTiV%2BYjHfQj60FD3rVaXmzT
soNxRhKJ72uIHVMGVQiAGgB%2BqAbSqfKHDQtVOmJJgkHug%3D%3D
&SignatureMethod=HmacSHA256
&Expires=2011-10-18T22%3A52%3A43PST
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&SignatureVersion=2
&Signature=Dqlp3Sd6ljTUA9Uf6SGtEExwUQEXAMPLE
```

Sample Response

```
<DeleteMessageBatchResponse>
    <DeleteMessageBatchResult>
        <DeleteMessageBatchResultEntry>
            <Id>msg1</Id>
        </DeleteMessageBatchResultEntry>
        <DeleteMessageBatchResultEntry>
            <Id>msg2</Id>
        </DeleteMessageBatchResultEntry>
    </DeleteMessageBatchResult>
    <ResponseMetadata>
        <RequestId>d6f86b7a-74d1-4439-b43f-196a1e29cd85</RequestId>
    </ResponseMetadata>
</DeleteMessageBatchResponse>
```

Related Actions

- SendMessage (p. 42)
- DeleteMessage (p. 19)
- ReceiveMessage (p. 36)

DeleteQueue

Description

The `DeleteQueue` action deletes the queue specified by the queue URL, regardless of whether the queue is empty. If the specified queue does not exist, SQS returns a successful response.

Caution

Use `DeleteQueue` with care; once you delete your 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 the 60 seconds, the queue and that message you sent no longer exist. Also, when you delete a queue, you must wait at least 60 seconds before creating a queue with the same name.

We reserve the right to delete queues that have had no activity for more than 30 days. For more information, see About SQS Queues in the Amazon SQS Developer Guide.

Request Parameters

The `DeleteQueue` action uses no special request parameters besides the common request parameters all actions use (for more information, see About SQS Queues in the Amazon SQS Developer Guide).

Response Elements

The response contains no special elements besides the common elements in all successful responses (for more information, see Responses section in the Amazon SQS Developer Guide.

Special Errors

The `DeleteQueue` action returns no special errors besides the common errors all actions return (for more information, see Common Errors (p. 54)).

Examples

The following example Query request deletes the specified queue.

Sample Request

```
http://sqs.us-east-1.amazonaws.com/123456789012/testQueue/
?Action=DeleteQueue
&Version=2009-02-01
&SignatureMethod=HmacSHA256
&Expires=2009-04-18T22%3A52%3A43PST
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&SignatureVersion=2
&Signature=Dqlp3Sd6ljTUA9Uf6SGtEExwUQEXAMPLE
```

Sample Response

```
<DeleteQueueResponse>
    <ResponseMetadata>
        <RequestId>
            6fde8d1e-52cd-4581-8cd9-c512f4c64223
        </RequestId>
    </ResponseMetadata>
</DeleteQueueResponse>
```

Related Actions

- CreateQueue (p. 15)
- ListQueues (p. 34)

GetQueueAttributes

Description

The `GetQueueAttributes` action returns one or all attributes of a queue.

Calling `GetQueueAttributes` with the *`AttributeName`* set to `All` returns values for each API version as indicated in the following table.

Attribute	2008-01-01	2009-02-01	2011-10-01
`VisibilityTimeout`	✓	✓	✓
`ApproximateNumberOfMessages`	✓	✓	✓
`ApproximateNumberOfMessagesNotVisible`		✓	✓
`CreatedTimestamp`		✓	✓
`LastModifiedTimestamp`		✓	✓
`QueueArn`		✓	✓
`MaximumMessageSize`		✓	✓
`MessageRetentionPeriod`		✓	✓
`Policy`		✓	✓
`DelaySeconds`			✓
`ApproximateNumberOfMessagesDelayed`			✓

Note

Going forward, new attributes might be added. If you are writing code that calls this action, we recommend that you structure your code so that it can handle new attributes gracefully.

Note

Some API operations take lists of parameters. These lists are specified using the `param.n` notation. Values of n are integers starting from 1. For example, a parameter list with two elements looks like this:

```
&Attribute.1=this
&Attribute.2=that
```

Request Parameters

The following table lists the special request parameters the `GetQueueAttributes` action uses in addition to the common request parameters all actions use (for more information, see Common Query Parameters (p. 53)).

Name	Description	Required
AttributeName.n	The attribute you want to get. • **All—**returns all values. • **ApproximateNumberOfMessages—**returns the approximate number of visible messages in a queue. For more information, see Resources Required to Process Messages in the Amazon SQS Developer Guide. • **ApproximateNumberOfMessagesNotVisible—**returns the approximate number of messages that are not timed-out and not deleted. For more information, see Resources Required to Process Messages in the Amazon SQS Developer Guide. • **ApproximateNumberOfMessagesDelayed—**returns the approximate number of messages that are not visible because you have set a positive delay value on the queue. • **VisibilityTimeout—**returns the visibility timeout for the queue. For more information about visibility timeout, see Visibility Timeout in the Amazon SQS Developer Guide. • **CreatedTimestamp—**returns the time when the queue was created (epoch time in seconds). • **LastModifiedTimestamp—**returns the time when the queue was last changed (epoch time in seconds). • **Policy—**returns the queue's policy. • **MaximumMessageSize—**returns the limit of how many bytes a message can contain before Amazon SQS rejects it. • **MessageRetentionPeriod—**returns the number of seconds Amazon SQS retains a message. • **QueueArn—**returns the queue's Amazon resource name (ARN). • **DelaySeconds—**The time in seconds that the delivery of all messages in the queue will be delayed. Type: String Valid values: `All | ApproximateNumberOfMessages | ApproximateNumberOfMessagesNotVisible | VisibilityTimeout | CreatedTimestamp | LastModifiedTimestamp | Policy | MaximumMessageSize | MessageRetentionPeriod | QueueArn | OldestMessageAge | DelaySeconds | ApproximateNumberOfMessagesDelayed` Default: None	Yes

Response Elements

The following table lists the response elements the `GetQueueAttributes` action returns in addition to the common response elements all actions return (for more information, see Responses section in the Amazon SQS Developer Guide.

Name	Description
`Attribute`	An element containing each attribute's information. Ancestor: `GetQueueAttributesResult` Children: • **Name—**The attribute name you requested. • **Value—**The value for the requested attribute. The `CreatedTimestamp` and the `LastModifiedTimestamp` are each returned as an integer representing the epoch time in seconds.

Special Errors

The following table lists the special errors the `GetQueueUrl` action returns in addition to the common errors all actions return (for more information, see Common Errors (p. 54)).

Error	Description	HTTP Status Code
`InvalidAttributeName`	Unknown attribute	400

Examples

The following example Query request gets all the attribute values for the specified queue.

Sample Request

```
http://sqs.us-east-1.amazonaws.com/123456789012/testQueue/
?Action=GetQueueAttributes
&AttributeName.1=All
&Version=2009-02-01
&SignatureMethod=HmacSHA256
&Expires=2009-04-18T22%3A52%3A43PST
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&SignatureVersion=2
&Signature=Dqlp3Sd6ljTUA9Uf6SGtEExwUQEXAMPLE
```

Sample Response

```
<GetQueueAttributesResponse>
  <GetQueueAttributesResult>
    <Attribute>
      <Name>VisibilityTimeout</Name>
      <Value>30</Value>
    </Attribute>
    <Attribute>
      <Name>ApproximateNumberOfMessages</Name>
      <Value>0</Value>
    </Attribute>
```

```
    <Attribute>
      <Name>ApproximateNumberOfMessagesNotVisible</Name>
      <Value>0</Value>
    </Attribute>
    <Attribute>
      <Name>CreatedTimestamp</Name>
      <Value>1286771522</Value>
    </Attribute>
    <Attribute>
      <Name>LastModifiedTimestamp</Name>
      <Value>1286771522</Value>
    </Attribute>
    <Attribute>
      <Name>QueueArn</Name>
      <Value>arn:aws:sqs:us-east-1:123456789012:qfoo</Value>
    </Attribute>
    <Attribute>
      <Name>MaximumMessageSize</Name>
      <Value>8192</Value>
    </Attribute>
    <Attribute>
      <Name>MessageRetentionPeriod</Name>
      <Value>345600</Value>
    </Attribute>
  </GetQueueAttributesResult>
  <ResponseMetadata>
    <RequestId>1ea71be5-b5a2-4f9d-b85a-945d8d08cd0b</RequestId>
  </ResponseMetadata>
</GetQueueAttributesResponse>
```

The following example Query request gets two attribute values for the specified queue.

Sample Request

```
http://sqs.us-east-1.amazonaws.com/123456789012/testQueue/
?Action=GetQueueAttributes
&Action=GetQueueAttributes
&Version=2011-10-01
&AttributeName.1=VisibilityTimeout
&AttributeName.2=DelaySeconds
&SignatureMethod=HmacSHA256
&Expires=2011-08-18T22%3A52%3A43PST
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&SignatureVersion=2
&Signature=Dqlp3Sd6ljTUA9Uf6SGtEExwUQEXAMPLE
```

Sample Response

```
<GetQueueAttributesResponse>
  <GetQueueAttributesResult>
    <Attribute>
      <Name>VisibilityTimeout</Name>
      <Value>30</Value>
    </Attribute>
    <Attribute>
```

```
      <Name>DelaySeconds</Name>
      <Value>0</Value>
    </Attribute>
</GetQueueAttributesResponse>
```

Related Actions

- SetQueueAttributes (p. 49)

GetQueueUrl

Description

The `GetQueueUrl` action returns the Uniform Resource Locater (URL) of a queue. This action provides a simple way to retrieve the URL of an SQS queue.

To access a queue that belongs to another AWS account, use the *`QueueOwnerAWSAccountId`* parameter to specify the account ID of the queue's owner. The queue's owner must grant you permission to access the queue. For more information about shared queue access, see AddPermission (p. 6) or go to Shared Queues in the *Amazon SQS Developer Guide*.

Request Parameters

The following table lists the special request parameters the `GetQueueUrl` action uses in addition to the common request parameters all actions use (for more information, see Common Query Parameters (p. 53)).

Name	Description	Required
`QueueName`	The name of an existing queue. Type: String Constraints: Maximum 80 characters; alphanumeric characters, hyphens (-), and underscores (_) are allowed.	Yes
`QueueOwnerAWSAccountId`	The AWS account ID of the account that created the queue. Type: String	No

Response Elements

The following table lists the response elements the `GetQueueUrl` action returns in addition to the common response elements all actions return (for more information, see Responses section in the Amazon SQS Developer Guide.

Name	Description
`QueueUrl`	The queue URL for the queue that you created. Type: String Ancestor: `GetQueueUrlResult`

Special Errors

A special error can result if you attempt to access a queue that doesn't exist in the named account or if you don't have permission to access the queue.

Examples

The following example Query request gets the URL for the specified queue.

Sample Request

```
http://sqs.us-east-1.amazonaws.com/
?Action=GetQueueUrl
&QueueName=testQueue
&Version=2011-10-01
&SignatureMethod=HmacSHA256
&Expires=2011-10-24T22%3A52%3A43PST
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&SignatureVersion=2
&Signature=Dqlp3Sd6ljTUA9Uf6SGtEExwUQEXAMPLE
```

Sample Response

```
<GetQueueUrlResponse>
    <GetQueueUrlResult>
        <QueueUrl>
            http://sqs.us-east-1.amazonaws.com/123456789012/testQueue
        </QueueUrl>
    </GetQueueUrlResult>
    <ResponseMetadata>
        <RequestId>470a6f13-2ed9-4181-ad8a-2fdea142988e</RequestId>
    </ResponseMetadata>
</GetQueueUrlResponse>
```

Related Actions

- CreateQueue (p. 15)
- ListQueues (p. 34)

ListQueues

Description

The `ListQueues` action returns a list of your queues. The maximum number of queues that can be returned is 1000. If you specify a value for the optional *`QueueNamePrefix`* parameter, only queues with a name beginning with the specified value are returned.

Request Parameters

The following table lists the special request parameters the `ListQueues` action uses in addition to the common request parameters all actions use (for more information, see Common Query Parameters (p. 53)).

Name	Description	Required
`QueueNamePrefix`	String to use for filtering the list results. Only those queues whose name begins with the specified string are returned. Type: String Constraints: Maximum 80 characters; alphanumeric characters, hyphens (-), and underscores (_) are allowed.	No

Response Elements

The following table lists the response elements the `ListQueues` action returns in addition to the common response elements all actions return (for more information, see Responses section in the Amazon SQS Developer Guide.

Name	Description
`QueueUrl`	The queue URL for a queue. The response can contain up to 1000 `QueueUrl` elements. Type: String Ancestor: `ListQueuesResult`

Special Errors

The `ListQueues` action returns no special errors besides the common errors all actions return (for more information, see Common Errors (p. 54)).

Examples

The following example Query request returns the queues whose names begin with the letter "T".

Sample Request

```
http://sqs.us-east-1.amazonaws.com/
?Action=ListQueues
&QueueNamePrefix=t
```

```
&Version=2009-02-01
&SignatureMethod=HmacSHA256
&Expires=2009-04-18T22%3A52%3A43PST
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&SignatureVersion=2
&Signature=Dqlp3Sd6ljTUA9Uf6SGtEExwUQEXAMPLE
```

Sample Response

```
<ListQueuesResponse>
    <ListQueuesResult>
        <QueueUrl>
            http://sqs.us-east-1.amazonaws.com/123456789012/testQueue
        </QueueUrl>
    </ListQueuesResult>
    <ResponseMetadata>
        <RequestId>
            725275ae-0b9b-4762-b238-436d7c65a1ac
        </RequestId>
    </ResponseMetadata>
</ListQueuesResponse>
```

Related Actions

- CreateQueue (p. 15)
- DeleteQueue (p. 24)

ReceiveMessage

Description

The `ReceiveMessage` action retrieves one or more messages from the specified queue. The `ReceiveMessage` action does not delete the message after it is retrieved. To delete a message, you must use the `DeleteMessage` action. For more information about message deletion in the message life cycle, see Message Lifecycle.

For each message returned, the response includes the following:

- Message body
- MD5 digest of the message body (for information about MD5, go to http://faqs.org/rfcs/rfc1321.html)
- Message ID you received when you sent the message to the queue
- Receipt handle

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).

Note

Due to the distributed nature of the queue, a weighted random set of machines is sampled on a `ReceiveMessage` call. That means only the messages on the sampled machines are returned. If the number of messages in the queue is small (less than 1000), it is likely you will 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; in which case you should repeat the request.

You can provide the *`VisibilityTimeout`* parameter in your request, which will be applied to the messages that SQS returns in the response. If you do not include the parameter, the overall visibility timeout for the queue is used for the returned messages. For more information, see Visibility Timeout in the Amazon SQS Developer Guide.

Note

Going forward, new attributes might be added. If you are writing code that calls this action, we recommend that you structure your code so that it can handle new attributes gracefully.

Note

Some API operations take lists of parameters. These lists are specified using the `param.n` notation. Values of n are integers starting from 1. For example, a parameter list with two elements looks like this:

```
&Attribute.1=this
&Attribute.2=that
```

Request Parameters

The following table lists the special request parameters the `ReceiveMessage` action uses in addition to the common request parameters all actions use (for more information, see Common Query Parameters (p. 53)).

Name	Description	Required
`AttributeName.n`	The attribute you want to get. • **All—**returns all values. • **SenderId—**returns the AWS account number (or the IP address, if anonymous access is allowed) of the sender. • **SentTimestamp—**returns the time when the message was sent (epoch time in milliseconds). • **ApproximateReceiveCount—**returns the number of times a message has been received but not deleted. • **ApproximateFirstReceiveTimestamp—**returns the time when the message was first received (epoch time in milliseconds). Type: String Valid values: `All | SenderId | SentTimestamp | ApproximateReceiveCount | ApproximateFirstReceiveTimestamp` Default: None	No
`MaxNumberOfMessages`	Maximum number of messages to return. SQS never returns more messages than this value but might return fewer. Not necessarily all the messages in the queue are returned (for more information, see the preceding note about machine sampling). Type: Integer from 1 to 10 Default: `1`	No
`VisibilityTimeout`	The duration (in seconds) that the received messages are hidden from subsequent retrieve requests after being retrieved by a `ReceiveMessage` request. Type: Integer Constraints: 0 to 43200 (maximum 12 hours) Default: The visibility timeout for the queue	No

Response Elements

The following table lists the response elements the `ReceiveMessage` action returns in addition to the common response elements all actions return (for more information, see Responses section in the Amazon SQS Developer Guide.

Name	Description
`Message`	An element containing the information about the message. Ancestor: `ReceiveMessageResult` Children: • **Body—**The message's contents (not URL encoded) • **MD5OfBody—**An MD5 digest of the non-URL-encoded message body string • **MessageId—**The message's SQS-assigned ID Length Constraint: Maximum 100 characters • **ReceiptHandle—**A string associated with a specific instance of receiving the message. Length Constraint: Maximum 1024 characters • **Attribute—**SenderId, SentTimestamp, ApproximateReceiveCount, and/or ApproximateFirstReceiveTimestamp. The SentTimestamp and ApproximateFirstReceiveTimestamp are each returned as an integer representing the epoch time in milliseconds.

Special Errors

The following table lists the special errors the `ReceiveMessage` action returns in addition to the common errors all actions return (for more information, see Common Errors (p. 54)).

Error	Description	HTTP Status Code
`ReadCountOutOfRange`	The value for *`MaxNumberOfMessages`* is not valid (must be from 1 to 10).	400

Examples

The following example Query request receives messages from the specified queue.

Sample Request

```
http://sqs.us-east-1.amazonaws.com/123456789012/testQueue/
?Action=ReceiveMessage
&MaxNumberOfMessages=5
&VisibilityTimeout=15
&AttributeName=All;
&Version=2009-02-01
&SignatureMethod=HmacSHA256
&Expires=2009-04-18T22%3A52%3A43PST
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&SignatureVersion=2
&Signature=Dqlp3Sd6ljTUA9Uf6SGtEExwUQEXAMPLE
```

Sample Response

```
<ReceiveMessageResponse>
  <ReceiveMessageResult>
    <Message>
      <MessageId>
        5fea7756-0ea4-451a-a703-a558b933e274
      </MessageId>
      <ReceiptHandle>
        MbZj6wDWli+JvwwJaBV+3dcjk2YW2vA3+STFFljTM8tJJg6HRG6PYSasuWXPJB+Cw
        Lj1FjgXUv1uSj1gUPAWV66FU/WeR4mq2OKpEGYWbnLmpRCJVAyeMjeU5ZBdtcQ+QE
        auMZc8ZRv37sIW2iJKq3M9MFx1YvV11A2x/KSbkJ0=
      </ReceiptHandle>
      <MD5OfBody>
        fafb00f5732ab283681e124bf8747ed1
      </MD5OfBody>
      <Body>This is a test message</Body>
      <Attribute>
        <Name>SenderId</Name>
        <Value>195004372649</Value>
      </Attribute>
      <Attribute>
        <Name>SentTimestamp</Name>
        <Value>1238099229000</Value>
      </Attribute>
      <Attribute>
        <Name>ApproximateReceiveCount</Name>
        <Value>5</Value>
      </Attribute>
      <Attribute>
        <Name>ApproximateFirstReceiveTimestamp</Name>
        <Value>1250700979248</Value>
      </Attribute>
    </Message>
  </ReceiveMessageResult>
  <ResponseMetadata>
    <RequestId>
      b6633655-283d-45b4-aee4-4e84e0ae6afa
    </RequestId>
  </ResponseMetadata>
</ReceiveMessageResponse>
```

Related Actions

- DeleteMessage (p. 19)
- SendMessage (p. 42)
- SetQueueAttributes (p. 49)

RemovePermission

Description

The `RemovePermission` action revokes any permissions in the queue policy that matches the *`Label`* parameter. Only the owner of the queue can remove permissions.

Request Parameters

The following table lists the special request parameters the action uses (in addition to the common request parameters listed in Common Query Parameters (p. 53)).

Name	Description	Required
`Label`	The identification of the permission you want to remove. This is the label you added in AddPermission (p. 6). Type: String (example: `AliceSendMessage`) Default: None	Yes

Response Elements

The response contains no special elements besides the common elements listed in Responses section in the Amazon SQS Developer Guide.

Special Errors

The action returns no errors besides the common errors listed in Common Errors (p. 54).

Examples

The following example Query request removes the *`testLabel`* permission on the queue named testQueue.

Sample Request

```
http://sqs.us-east-1.amazonaws.com/123456789012/testQueue/
?Action=RemovePermission
&Label=testLabel
&Version=2009-02-01
&SignatureMethod=HmacSHA256
&Expires=2009-04-18T22%3A52%3A43PST
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&SignatureVersion=2
&Signature=Dqlp3Sd6ljTUA9Uf6SGtEExwUQEXAMPLE
```

Sample Response

```
<RemovePermissionResponse>
    <ResponseMetadata>
        <RequestId>
```

```
            f8bdb362-6616-42c0-977a-ce9a8bcce3bb
        </RequestId>
    </ResponseMetadata>
</RemovePermissionResponse>
```

Related Actions

- AddPermission (p. 6)

SendMessage

Description

The `SendMessage` action delivers a message to the specified queue. The maximum allowed message size is 64 KB.

Important

The following list shows the characters (in Unicode) allowed in your message, according to the W3C XML specification (for more information, go to http://www.w3.org/TR/REC-xml/#charsets). If you send any characters not included in the list, your request will be rejected.

#x9 | #xA | #xD | [#x20 to #xD7FF] | [#xE000 to #xFFFD] | [#x10000 to #x10FFFF]

Request Parameters

The following table lists the special request parameters the `SendMessage` action uses in addition to the common request parameters all actions use (for more information, see Common Query Parameters (p. 53)).

Name	Description	Required
`MessageBody`	The message to send. Type: String maximum 64 KB in size. For a list of allowed characters, see the preceding important note.	Yes
`DelaySeconds`	The number of seconds to delay a specific message. Messages with a positive *`DelaySeconds`* value become available for processing after the delay time is finished. If you don't specify a value, the default value for the queue applies. Type: Integer from 0 to 900 (15 minutes). If this parameter is not used, the default value for the queue applies.	No

Response Elements

The following table lists the response elements the `SendMessage` action returns in addition to the common response elements all actions return (for more information, see Responses section in the Amazon SQS Developer Guide.

Name	Description
`MD5OfMessageBody`	An MD5 digest of the non-URL-encoded message body string. You can use this to verify that SQS received the message correctly. SQS first URL decodes the message before creating the MD5 digest. For information about MD5, go to http://faqs.org/rfcs/rfc1321.html. Type: String Ancestor: `SendMessageResult`

Name	Description
MessageId	An element containing the message ID of the message sent to the queue. For more information, see Queue and Message Identifiers in the Amazon SQS Developer Guide. Type: String Length Constraint: Maximum 100 characters Ancestor: SendMessageResult

Special Errors

The following table lists the special errors the SendMessage action returns in addition to the common errors all actions return (for more information, see Common Errors (p. 54)).

Error	Description	HTTP Status Code
InvalidMessageContents	The message contains characters outside the allowed set.	400
MessageTooLong	The message size cannot exceed 64 KB.	400

Examples

Sample Request

The following example SendMessage request sends a message containing "Your Message Text" to the queue. You must URL encode the entire URL; however, we've URL encoded only the message body to make the example easier for you to read.

```
http://sqs.us-east-1.amazonaws.com/123456789012/testQueue/
?Action=SendMessage
&MessageBody=This+is+a+test+message
&Version=2009-02-01
&SignatureMethod=HmacSHA256
&Expires=2009-04-18T22%3A52%3A43PST
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&SignatureVersion=2
&Signature=Dqlp3Sd6ljTUA9Uf6SGtEExwUQEXAMPLE
```

Sample Response

The following example response includes the MD5 digest for Your Message Text.

```
<SendMessageResponse>
    <SendMessageResult>
        <MD5OfMessageBody>
            fafb00f5732ab283681e124bf8747ed1
        </MD5OfMessageBody>
        <MessageId>
```

```
            5fea7756-0ea4-451a-a703-a558b933e274
        </MessageId>
    </SendMessageResult>
    <ResponseMetadata>
        <RequestId>
            27daac76-34dd-47df-bd01-1f6e873584a0
        </RequestId>
    </ResponseMetadata>
</SendMessageResponse>
```

Related Actions

- DeleteMessage (p. 19)
- ReceiveMessage (p. 36)

SendMessageBatch

Description

The `SendMessageBatch` action delivers up to ten messages to the specified queue. The maximum allowed individual message size is 64 KiB (65,536 bytes).

The maximum total payload size (i.e., the sum of all a batch's individual message lengths) is also 64 KiB (65,536 bytes).

If the *`DelaySeconds`* parameter is not specified for an entry, the default for the queue is used.

Important

The following list shows the characters (in Unicode) that are allowed in your message, according to the W3C XML specification (for more information, go to http://www.w3.org/TR/REC-xml/#charsets). If you send any characters that are not included in the list, your request will be rejected.

#x9 | #xA | #xD | [#x20 to #xD7FF] | [#xE000 to #xFFFD] | [#x10000 to #x10FFFF]

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.

Note

Some API operations take lists of parameters. These lists are specified using the `param.n` notation. Values of n are integers starting from 1. For example, a parameter list with two elements looks like this:

```
&Attribute.1=this
&Attribute.2=that
```

Request Parameters

The following table lists the special request parameters that the `SendMessageBatch` action uses in addition to the common request parameters that all actions use (for more information, see Common Query Parameters (p. 53)).

Name	Description	Required
`SendMessageBatchRequestEntry.n.Id`	An identifier that you assign to the message. Requests that are part of the same call to `SendMessageBatch` cannot have the same identifier. Type: String.	Yes

Name	Description	Required
SendMessageBatchRequestEntry.n.MessageBody	The message to send. Type: String maximum 64 KiB in size. For a list of allowed characters, see the preceding important note.	Yes
SendMessageBatchRequestEntry.n.DelaySeconds	The number of seconds to delay a specific message. Messages with a positive *DelaySeconds* value become available for processing after the delay time is finished. If you don't specify a value, the default value for the queue applies. Type: Integer from 0 to 900 (maximum 15 minutes).	No

Response Elements

For each message in the batch, the response contains a *SendMessageBatchResultEntry* tag if the message succeeds or a *BatchResultErrorEntry* tag if the message fails.

The following table lists the response elements the SendMessageBatch action returns in addition to the common response elements all actions return (for more information, see Responses section in the Amazon SQS Developer Guide.

Name	Description
SendMessageBatchResultEntry	An element containing a successfully enqueued message's information. Ancestor: SendMessageBatchResult Children: • **Id—**The Id name that you assigned to the message. • **MessageId—**An identifier that is assigned by Amazon SQS. Length Constraint: Maximum 100 characters • **MD5OfMessageBody—**An MD5 digest of the message body.
BatchResultErrorEntry	An element containing information about a message that was not added to the queue. Ancestor: SendMessageBatchResult Children: • **Id—**The Id name that you assigned to the message. • **SenderFault—**A boolean value. • **Code—**A short string description of the error. • **Message—**A description of the error.

Special Errors

Because batch requests can result in a combination of successful and unsuccessful actions, an HTTP status code of 200 might include an error that is described in this section.

The following table lists the special errors that the `SendMessageBatch` action returns in addition to the common errors that all actions return (for more information, see Common Errors (p. 54)).

Error	Description
`InvalidMessageContents`	The message contains characters outside the allowed set.
`MessageTooLong`	The message size cannot exceed 64 KB.
`BatchResultErrorEntry`	A message was not added to the queue. For more information, see Response Elements (p. 46).

Examples

Sample Request

The following example `SendMessageBatch` request sends a message containing "Your Message Text" to the queue. You must URL encode the entire URL; however, we've URL encoded only the message body to make the example easier for you to read.

```
http://sqs.us-east-1.amazonaws.com/123456789012/testQueue/
?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
&Version=2011-10-01
&SignatureMethod=HmacSHA256
&Expires=2011-10-18T22%3A52%3A43PST
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&SignatureVersion=2
&Signature=Dqlp3Sd6ljTUA9Uf6SGtEExwUQEXAMPLE
```

Sample Response

The following example response includes the MD5 digest for `Your Message Text`.

```
<SendMessageBatchResponse>
<SendMessageBatchResult>
    <SendMessageBatchResultEntry>
        <Id>test_msg_001</Id>
        <MessageId>0a5231c7-8bff-4955-be2e-8dc7c50a25fa</MessageId>
        <MD5OfMessageBody>0e024d309850c78cba5eabbeff7cae71</MD5OfMessageBody>
    </SendMessageBatchResultEntry>
    <SendMessageBatchResultEntry>
        <Id>test_msg_002</Id>
        <MessageId>15ee1ed3-87e7-40c1-bdaa-2e49968ea7e9</MessageId>
```

```
        <MD5OfMessageBody>7fb8146a82f95e0af155278f406862c2</MD5OfMessageBody>
    </SendMessageBatchResultEntry>
</SendMessageBatchResult>
<ResponseMetadata>
    <RequestId>ca1ad5d0-8271-408b-8d0f-1351bf547e74</RequestId>
</ResponseMetadata>
</SendMessageBatchResponse>
```

Related Actions

- SendMessage (p. 42)
- DeleteMessage (p. 19)
- ReceiveMessage (p. 36)

SetQueueAttributes

Description

The `SetQueueAttributes` action sets one attribute of a queue per request. When you change a queue's attributes, the change can take up to 60 seconds to propagate throughout the SQS system.

Request Parameters

The following table lists the special request parameters the `SetQueueAttributes` action uses in addition to the common request parameters all actions use (for more information, see Common Query Parameters (p. 53)).

Name	Description	Required
`Attribute.Name`	The name of the attribute you want to set. • **VisibilityTimeout** The length of time (in seconds) that a message received from a queue will be invisible to other receiving components when they ask to receive messages. For more information about `VisibilityTimeout`, see Visibility Timeout in the Amazon SQS Developer Guide. • **Policy** The formal description of the permissions for a resource. For more information about `Policy`, see Basic Policy Structure in the Amazon SQS Developer Guide. • **MaximumMessageSize** The limit of how many bytes a message can contain before Amazon SQS rejects it. • **MessageRetentionPeriod** The number of seconds Amazon SQS retains a message. • **DelaySeconds** The time in seconds that the delivery of all messages in the queue will be delayed. Type: String Valid Values: `VisibilityTimeout | Policy | MaximumMessageSize | MessageRetentionPeriod | DelaySeconds` Default: None	Yes

Name	Description	Required
Attribute.Value	The value of the attribute you want to set. To delete a queue's access control policy, set the policy to "". Constraints: Constraints are specific for each value. • **VisibilityTimeout** An integer from 0 to 43200 (12 hours). The default for this attribute is 30 seconds. • **Policy** A valid form-url-encoded policy. For more information about policy structure, see Basic Policy Structure in the Amazon SQS Developer Guide. For more information about form-url-encoding, see http://www.w3.org/MarkUp/html-spec/html-spec_8.html#SEC8.2.1. • **MaximumMessageSize** An integer from 1024 bytes (1 KiB) up to 65536 bytes (64 KiB). The default for this attribute is 65536 (64 KiB). • **MessageRetentionPeriod** Integer representing seconds, from 60 (1 minute) to 1209600 (14 days). The default for this attribute is 345600 (4 days). • **DelaySeconds** An integer from 0 to 900 (15 minutes). The default for this attribute is 0. Default: Varies according to attribute	Yes

Response Elements

The response contains no special elements besides the common elements in all successful responses (for more information, see Responses section in the Amazon SQS Developer Guide.

Special Errors

The following table lists the special errors the `SetQueueAttributes` action returns in addition to the common errors all actions return (for more information, see Common Errors (p. 54)).

Error	Description	HTTP Status Code
`InvalidAttributeName`	Unknown attribute	400

Examples

Sample Request with Policy

The following example Query request sets a policy that gives all users ReceiveMessage permission for the queue named testQueue.

```
http://sqs.us-east-1.amazonaws.com/123456789012/testQueue/
?Action=SetQueueAttributes
&Version=2009-02-01
```

```
&Attribute.Name=Policy
&Attribute.Value=%7B%22Version%22%3A%222008-10-17%22%2C%22Id%22
%3A%22%2F123456789012%2FtestQueue%2FSQSDefaultPolicy%22%2C%22Stat
ement%22%3A%5B%7B%22Sid%22%3A%22Queue1ReceiveMessage%22%2C%22Effe
ct%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%3Aaws%3Asqs%3Aus%2Deast%2D1%3A123456789012%3AtestQueue%22%7D%5D%7D
&Timestamp=2009-05-06T16%3A57%3A31.000Z
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&Signature=%2Bd7ZlPIdnb%2BhpEna2TgfwQjfGF8%3D
```

The above policy value is form-url-encoded from the following text:

```
{
    "Version":"2008-10-17",
    "Id":"/123456789012/testQueue/SQSDefaultPolicy",
    "Statement":  [
        {
        "Sid":"Queue1ReceiveMessage",
        "Effect":"Allow",
        "Principal":{"AWS":"*"},
        "Action":"SQS:ReceiveMessage",
        "Resource":"arn:aws:sqs:us-east-1:123456789012:testQueue"
        }
    ]
}
```

For more examples of policies, see Amazon SQS Policy Examples in the Amazon SQS Developer Guide.

Sample Request with VisibilityTimeout

The following example Query request sets the visibility timeout to 35 seconds for the queue named 123456789012/testQueue.

```
http://sqs.us-east-1.amazonaws.com/123456789012/testQueue/
?Action=SetQueueAttributes
&Attribute.Name=VisibilityTimeout
&Attribute.Value=35
&Version=2009-02-01
&SignatureMethod=HmacSHA256
&Expires=2009-04-18T22%3A52%3A43PST
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&SignatureVersion=2
&Signature=Dqlp3Sd6ljTUA9Uf6SGtEExwUQEXAMPLE
```

Sample Response

```
<SetQueueAttributesResponse>
    <ResponseMetadata>
        <RequestId>
            e5cca473-4fc0-4198-a451-8abb94d02c75
        </RequestId>
    </ResponseMetadata>
</SetQueueAttributesResponse>
```

Related Actions

- GetQueueAttributes (p. 26)

Common Query Parameters

The following table lists common parameters used by Query requests to SQS.

Parameter Name	Description
Action	The action to perform. For example: `CreateQueue`..
AWSAccessKeyId	Your Access Key ID. For example: `AKIAIOSFODNN7EXAMPLE`. For more information, see Your AWS Identifiers in the Amazon SQS Developer Guide.
Expires	The date and time at which the signature included in the request expires, in the format YYYY-MM-DDThh:mm:ssZ, as specified in the ISO 8601 standard. Query requests must include either *Timestamp* or *Expires*, but not both.
Signature	A request signature (for information, see Request Authentication in the Amazon SQS Developer Guide). For example: `Qnpl4Qk/7tINHzfXCiT7VbBatDA=`.
SignatureMethod	Required when you use signature version 2 with Query requests. For more information, see Query Request Authentication in the Amazon SQS Developer Guide.
SignatureVersion	For more information, see Query Request Authentication in the Amazon SQS Developer Guide.
Timestamp	The date and time the request is signed, in the format YYYY-MM-DDThh:mm:ssZ, as specified in the ISO 8601 standard. Query requests must include either *Timestamp* or *Expires*, but not both.
Version	The API version to use, as specified in the WSDL. For example: 2011-10-01.

For Query requests, parameter values must be URL-encoded. This is true for any Query parameter passed to SQS and is typically necessary in the *Signature*, *ReceiptHandle*, and *MessageBody* parameters. Some clients do this automatically, but this is not the norm.

Common Errors

Topics

This section lists the errors shared by all actions. Errors specific to an action are listed in the topic about that action. For information about the structure of an error response, see Responses section in the Amazon SQS Developer Guide.

Important

We might throttle requests to Amazon SQS as necessary. When we throttle, we return a 503 (service unavailable) HTTP status code. Your system should be prepared to retry any request that receives a 503 code.

List of Errors

Error	Description	HTTP Status Code
`AccessDenied`	Access to the resource is denied.	403
`AuthFailure`	A value used for authentication could not be validated, such as *`Signature`*. For an example, see Example Response for AuthFailure Error (p. 56).	401
`AWS.SimpleQueueService.InternalError`	There is an internal problem with SQS, which you cannot resolve. Retry the request. If the problem persists, contact us through the Amazon SQS Discussion Forums.	500
`AWS.SimpleQueueService.NonExistentQueue`	Queue does not exist.	400

Error	Description	HTTP Status Code
`ConflictingQueryParameter`	The query parameter <*`parameter`*> is invalid. Its structure conflicts with that of another parameter.	400
`InternalError`	There is an internal problem with SQS, which you cannot resolve. Retry the request. If the problem persists, contact us through the Amazon SQS Discussion Forums.	500
`InvalidAccessKeyId`	AWS was not able to validate the provided access credentials.	401
`InvalidAction`	The action specified was invalid.	400
`InvalidAddress`	The address <*`address`*> is not valid for this web service.	404
`InvalidHttpRequest`	Invalid HTTP request. Reason: <*`reason`*>.	400
`InvalidParameterCombination`	Two parameters were specified that cannot be used together, such as *`Timestamp`* and *`Expires`*.	400
`InvalidParameterValue`	One or more parameters cannot be validated.	400
`InvalidQueryParameter`	The query parameter <*`parameter`*> is invalid. Please see service documentation for correct syntax.	400
`InvalidRequest`	The service cannot handle the request. Request is invalid.	400
`InvalidSecurity`	The provided security credentials are not valid. Reason: <*`reason`*>.	403
`InvalidSecurityToken`	The security token used in the request is invalid. Reason: <*`reason`*>.	400
`MalformedVersion`	Version not well formed: <*`version`*>. Must be in YYYY-MM-DD format.	400
`MissingClientTokenId`	Request must contain AWSAccessKeyId or X.509 certificate.	403
`MissingCredentials`	AWS was not able to authenticate the request: access credentials are missing.	401
`MissingParameter`	A required parameter is missing.	400
`NoSuchVersion`	An incorrect version was specified in the request.	400

Error	Description	HTTP Status Code
`NotAuthorizedToUseVersion`	Users who sign up to use Amazon SQS after February 1, 2008, must use API version 2008-01-01 and above; not previous API versions.	401
`RequestExpired`	The timestamp used with the signature has expired.	400
`RequestThrottled`	Request is throttled.	503
`ServiceUnavailable`	A required server needed by SQS is unavailable. This error is often temporary; resend the request after a short wait.	503
`X509ParseError`	Could not parse X.509 certificate.	400

Example Response for AuthFailure Error

The following sample response is for an `AuthFailure` error.

```
<ErrorResponse>
  <Error>
    <Type>
      Sender
    </Type>
    <Code>
      AuthFailure
    </Code>
    <Message>
      The provided signature is not valid for this access token
    </Message>
    <Detail/>
  </Error>
  <RequestId>
     ef3aba6a-dc84-4937-91bf-cef2ddd6775a
  </RequestId>
</ErrorResponse>
```

Amazon SQS Resources

The following table lists related resources that you'll find useful as you work with this service.

Resource	Description
Amazon Simple Queue Service Getting Started Guide	The getting started guide provides a quick tutorial of the service based on a simple use case. Examples and instructions in multiple programming languages are included.
Amazon Simple Queue Service Developer Guide	The developer guide provides a detailed discussion of the service. It includes an architectural overview and a programming reference
Amazon SQS Release Notes	The release notes give a high-level overview of the current release. They specifically note any new features, corrections, and known issues.
AWS Developer Resource Center	A central starting point to find documentation, code samples, release notes, and other information to help you build innovative applications with AWS.
Discussion Forums	A community-based forum for developers to discuss technical questions related to Amazon SQS.
AWS Support Center	The home page for AWS Technical Support, including access to our Developer Forums, Technical FAQs, Service Status page, and AWS Premium Support (if you are subscribed to this program).
AWS Premium Support Information	The primary web page for information about AWS Premium Support, a one-on-one, fast-response support channel to help you build and run applications on AWS Infrastructure Services.
Product information for Amazon SQS	The primary web page for information about Amazon SQS.
Contact Us	A central contact point for inquiries concerning AWS billing, account, events, abuse etc.
Conditions of Use	Detailed information about the copyright and trademark usage at Amazon.com and other topics.

Glossary

action	The action is the activity the principal has permission to perform. The action is B in the statement "A has permission to do B to C where D applies." The action is just the operation in the request to SQS. For example, Jane sends a request to Amazon SQS with `Action=ReceiveMessage`. For more information, see Shared Queues in the Amazon SQS Developer Guide.
conditions	The conditions are any restrictions or details about the permission. The condition is D in the statement "A has permission to do B to C where D applies." Following are some of the common types of conditions:
issuer	The issuer is the person who writes a policy to grant permissions to a resource. The issuer (by definition) is always the resource owner. AWS does not permit SQS users to create policies for resources they don't own. If John is the resource owner, AWS authenticates John's identity when he submits the policy he's written to grant permissions for that resource.
message ID	An identifier you get when you send a message to the queue.
permission	A permission allows or disallows access to a particular resource. You can state any permission like this: "A has permission to do B to C where D applies." For example, Jane (A) has permission to read messages (B) from John's Amazon SQS queue (C), as long as she asks to receive only a maximum of 10 messages from the queue at a time (D). Whenever Jane sends a request to Amazon SQS to use John's queue, the service checks to see if she has permission and if the request satisfies the conditions John set forth in the permission. For more information, see Shared Queues in the Amazon SQS Developer Guide.
queue URL	The URL uniquely identifying a queue.
policy	A policy is the formal description of the permissions for a resource. The Access Policy Language distinguishes between a policy and a statement. A policy is the complete document that can contain many different permissions for a given resource. A statement is the description of an individual permission. Therefore a policy can contain multiple statements. For example, a policy could specify that Jane can use John's queue (one statement), and Bob cannot use John's queue (another statement).
principal	The principal is the person or persons who receive the permission in the policy. The principal is A in the statement "A has permission to do B to C where D applies." In a policy, you may set the principal to "anyone" (i.e., you can specify a wildcard to represent all people). You might do this, for example, if you don't

	want to restrict access based on the actual identity of the requester, but instead on some other identifying characteristic such as the requester's IP address.
Query	This is a type of HTTP request that generally uses only the GET or POST HTTP method and a query string with parameters.
receipt handle	An identifier you get when you receive a message from the queue. You must provide this identifier when deleting a message from the queue or when changing a message's visibility timeout.
requester	The requester is the person who sends a request to an AWS service and asks for access to a particular resource. The requester sends a request to AWS that essentially says: "Can A do B to C where D applies?" In this question, the requester is A.
resource	The resource is the object the principal is requesting access to. The resource is C in the statement "A has permission to do B to C where D applies."
Secret Access Key	A key that Amazon Web Services (AWS) assigns to you when you sign up for an AWS account. Used for request authentication. For more information, see Your AWS Account in the Amazon SQS Developer Guide.
visibility timeout	The length of time (in seconds) that a message that has been received from a queue will be invisible to other receiving components when they ask to receive messages. During the visibility timeout, the component that received the message usually processes the message and then deletes it from the queue. For more information, see Visibility Timeout in the Amazon SQS Developer Guide.

Document History

This documentation is associated with the 2011-10-01 release of the Amazon Simple Queue Service. This guide was last updated on 25 June 2012.

The following table describes the important changes since the last release of the *Amazon Simple Queue Service API Reference*.

Change	Description	Release Date
New feature	The 2011-10-01 API version of Amazon SQS supports three new batch API actions. You can send or delete up to ten messages with a single call to either `SendMessageBatch` or `DeleteMessageBatch`. You can also change the visibility timeout of up to ten messages with one call to `ChangeMessageVisibilityBatch`. For more information about the new batch actions, see SendMessageBatch (p. 45), DeleteMessageBatch (p. 21), and ChangeMessageVisibilityBatch (p. 11).	In this release
New feature	The 2011-10-01 API version of Amazon SQS introduces a new `GetQueueUrl` action. You can quickly retrieve a queue's URL with `GetQueueUrl`. For more information about `GetQueueUrl`, see GetQueueUrl (p. 32).	In this release
New feature	The 2011-10-01 API version of Amazon SQS adds delay queues and message timers. Delay queues allow you to postpone the delivery of all messages in a queue for a specific number of seconds. Message timers allow you to postpone delivery of a single message for a specific number of seconds. For both delay queues and message timers, the delay period begins when you add the message to the queue. For more information about delay queues and message timers, go to Amazon SQS Delay Queues and Amazon SQS Message Timers in the *Amazon Simple Queue Service Developer Guide*.	In this release

Index