Amazon Simple Queue Service

Developer Guide (API Version 2006-04-01)

Amazon Simple Queue Service: Developer Guide (API Version 2006-04-01)

Published 2006-July-12

Third Party Information: This guide contains links to third-party websites that are not under the control of Amazon.com, and Amazon.com is not responsible for the content of any linked site. If you access a third-party website mentioned in this guide, then you do so at your own risk. Amazon.com provides these links at your own convenience, and the inclusion of the link does not imply that Amazon.com endorses or accepts any responsibility for the content on those third-party sites.

Table of Contents

Welcome to Amazon SQS

Amazon SQS™ is a distributed message queue system that provides a means for web service applications to quickly and reliably queue messages generated by one component to be consumed by another component. Using Amazon SQS, developers can decouple components of an application so that they run independently, with Amazon SQS easing messaging management between components. Any component of a distributed application can store any type of data in a fail-safe queue on Amazon.com. Any other component can then later retrieve the data programatically using the SQS API in SOAP, REST, or HTTP Query.

The queue acts as a buffer between the component producing and saving data, and the component retrieving the data for processing. Thus, the queue resolves issues that would otherwise arise if the producer were producing work faster than the consumer can process the work, or if the producer or consumer were only intermittently connected to the network.

SQS ensures "at least once" delivery of messages, and supports multiple readers and multiple writers interacting with the same queue. A single queue can be used simultaneously by many distributed application components, with no need for those components to coordinate with each other to share the queue.

The intended audience for the Amazon SQS is software developers who build distributed web-enabled applications. An application could typically use SQS as a buffer to manage the flow of data from one application component to another component at a different network location.

Amazon SQS Features

- At least once delivery of messages via a redundant infrastructure.
- Multiple writers and readers of the same queue.
- Highly concurrent access to messages.
- High availability for sending/retrieving messages.
- Messages size of 1 byte to 256K.
- Message are guaranteed to be available for 15 days after being sent.
- Settings configurable on a per queue basis.

Signing Up for SQS

Before you can begin using SQS, you must sign up to use the service. To sig up for SQS, you need to have an Amazon.com account and an AWS account. If you do not have these account, both will be created for you as part of the sign up process. The login information used is the same for both AWS accounts and Amazon.com accounts.

When you sign up for AWS, you immediately have access to all the free services offered by AWS. For information about which services are included, please see the Amazon Web Services web site.

To sign up to use SQS, please go to the Amazon SQS page, and then click *`Sign Up for Web Service`*, found in the top right corner of the page.

Follow the instructions displayed, after which you will receive confirmation that you have successfully signed up for the SQS service.

Amazon SQS is not free, and minimal charges will be incurred for usage of the service.

Life cycle of an SQS Message

A SQS message goes through the following steps in its lifecycle, from creation to deletion. It is assumed that a queue has already been created.

1. A message is created and sent to the queue using an action on the MessageQueue endpoint. The MessageQueue endpoint is represented by the queueURL returned from the CreateQueue or POST action that created the queue. The `SendMessage` action is used in Query and SOAP requests. The PUT action is used in REST requests.
2. When the consuming component is ready to process the message, it is retrieved from the queue by. This is done using an action on the MessageQueue endpoint, which is a Queue URL returned from the action used to create the queue. The `ReceiveMessage` action is used in Query and SOAP requests. The GET action is used in REST requests.

   While the receiver processes a message, it remains in the queue. However, it remains hidden and is not returned to subsequent `ReceiveMessasge` or `GET` request until the duration of the Visibility Timeout has expired.
3. The message is deleted from the queue. Messages remain in the queue until deleted. This is done using the `DeleteMessage` action in Query and SOAP requests. The DELETE action is used in REST requests. If the message is not deleted from the queue it will be returned by a `ReceiveMessage` request that occurs after the visibility timeout has expired, causing the message to be received more than once.

AWS Request Authentication

Request authentication is the process of verifying the identity of the sender of a request. In the context of Amazon Web Services (AWS) requests, authentication is the process by which AWS can verify that a request came from a registered user, as well as verify the identity of that registered user.

To enable authentication, each request must carry information about the identity of the request sender. The request must also contain additional information that AWS can use to verify that the request can only have been produced by the sender identified in the request. If the request passes this verification test it is determined to be “authentic” and AWS has sufficient information to verify the identity of the sender.

Verifying the identity of the sender of a request is important, as it ensures that only those requests made by the person or party responsible for the AWS account specified in the request are accepted and allowed to interact with AWS services. In this manner, request authentication allows Amazon to track the usage of AWS services on a per request basis. This enables Amazon to charge and bill AWS subscribers for use of AWS paid (not free) services. Authentication identifiers are also used for authorization, or access control. Users of AWS Web services can set access control policies on service resources, such as objects stored with Amazon S3. An access control policy is a set of permissions that allow or restrict access to a resource.

AWS Accounts

To access Amazon web services, a developer must create an AWS account. AWS accounts are associated with Amazon.com accounts. To sign in to an AWS account, a developer uses his or her Amazon.com account e-mail and password.

AWS Access Identifiers

AWS Access Identifiers are values assigned to, or associated with, an AWS account, and known only by the person responsible for that account. The identifiers are used by the developer in requests sent to AWS for the purposes of authentication.

Each AWS account is assigned an Access Key ID. The Access Key ID represents the symmetric access key (Access Key identifiers) for use with HMAC-SHA1 authentication.

Each AWS developer can alternatively use an X.509 certificate, which is self-identifying, for use with RSA-SHA1 authentication. However, certificates generated by AWS can be used only for authentication with AWS.

Note

Note: The developer's Access Key ID is also used to determine the AWS account for usage tracking billing. Access Identifiers should not be shared, as the AWS account associated with the Access Key provided in a request will be billed for that usage.

Access Key Identifiers

Upon creating an AWS account, the developer is assigned an Access Key ID (AWSAccessKeyId) and a Secret Access Key. The Access Key ID, which is associated with the AWS account, is used in requests to identify the party responsible for the request. However, because an Access Key ID is sent as a request parameter, it is not secret and could be used by anyone sending a request to AWS.

To protect from impersonation, the request sender must provide additional information that can be used to verify the sender's identity and ensure that the request is legitimate. This additional information, a request signature that is an HMAC-SHA1 hash calculated using the Secret Access Key, demonstrates possession of a shared secret known only to AWS and the sender of the request. A Secret Access Key is a 40-character alphanumeric sequence generated by AWS.

Note

Access Key Identifiers can be used to calculate HMAC-SHA1 request signatures in Query, REST, and SOAP requests.

X.509 Certidicate Identifiers

In SQS, a developer can choose to use X.509 certificates for authentication with SOAP requests rather than Access Key identifiers. X.509 certificates can be used only with SOAP. The certificate used can be one the developer already has, or a developer can generate a new certificate via the AWS portal.

AWS uses X.509 certificates only as a convenient way to carry a public key. Rather than trusting a certificate authority, AWS relies on your login to the developer portal to authenticate you and bind you to the certificate. This allows you to avoid the expense and effort required to obtain a certificate from a commercial certificate authority. For this reason, we recommend that you use a non-expiring, self-signed X.509 certificate for AWS.

When you upload your own certificate, AWS confirms that the certificate has not expired. AWS does not determine if the certificate has been revoked, either by checking a certificate revocation list (CRL) or by any other means. AWS does not validate the certificate with a certificate authority or any trusted third parties and does not validate the chain of signing authorities contained in the certificate. AWS does not use or validate the distinguished name or other identifiers contained within the certificate.

Once you have uploaded your certificate, AWS performs no further checks when you use it. Certificate revocation or expiration will not invalidate your certificate for AWS use. To invalidate your certificate for AWS use, you must delete the certificate or replace it with a new certificate, either one you upload or one generated for you by AWS.

When using a certificate to authenticate SOAP requests, the following WS-Security standards must be observed:

- http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3
- http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509PKIPathv1

- ValueType should be one of the following:
- The EncodingType must be "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary". If it is different, the request will be rejected with a WS-Security InvalidSecurityToken fault.

About X.509 Certificates Generated by AWS

Certificates generated via the AWS portal consist of two files: a certificate file and a private key file. The certificate files contain only the information necessary to use the certificate to generate signatures for requests to AWS. They do not contain any personal or company information, and are not registered with a Certificate Authority.

The contents of the generated certificate files are as follows:

Certificate File, cert-XXX.pem (where XXXX represents the certificate value)

```
-----BEGIN CERTIFICATE-----
<Base64 encoded DER certificate body>
-----END CERTIFICATE-----
```

Private Key file, pk-XXX.pem

```
-----BEGIN PRIVATE KEY-----
<Base64 encoded PKCS#8 private key>
-----END PRIVATE KEY-----
```

Once certificates have been downloaded, you should store them to use with your IDE or toolkit as desired.

Summary of HMAC-SHA-1 Request Authentication

The following steps are the basic steps used in authenticating requests to AWS. It is assumed that the developer has already registered with AWS and received an Access Key ID and Secret Access Key.

1. The sender constructs a request to AWS.
2. The sender calculates the request signature, a Keyed-Hashing for Message Authentication code (HMAC) with an SHA-1 hash function, as defined in the next section of this topic.
3. The sender of the request sends the request data, the signature, and Access Key ID (the key-identifier of the Secret Access Key used) to AWS.
4. AWS uses the Access Key ID to look up the Secret Access Key.
5. AWS generates a signature from the request data and the Secret Access Key using the same algorithm used to calculate the signature in the request.
6. If the signature generated by AWS matches the one sent in the request, the request is considered to be authentic. If the comparison fails, the request is discarded, and AWS returns an error response.

Calculating HMAC-SHA1 Request Signatures

Every request to AWS for which authentication is required must contain a request signature. A request signature is calculated by concatenating the values of request parameters, in alphabetical order, and then calculating an RFC 2104-compliant HMAC-SHA1 hash, using the Secret AWS Access Key as the key. The computed HMAC-SHA1 hash is passed in the Signature request parameter. For more information, see http://www.faqs.org/rfcs/rfc2104.html.

When a request is received, AWS verifies that the request signature is valid by computing an HMAC-SHA1 hash for the request, and then comparing the value of that hash with the value in the included in the request. If the values match, the identity of the sender is verified and the request is accepted. If the values do not match, the request is rejected.

> **Note**
>
> If a request contains a Timestamp parameter, the signature calculated for the request expires 15 minutes after the Timestamp value. If a request contains an Expires parameter, the signature expires at the time specified as the value for the Expired parameter.

The following steps demonstrate how to calculate a signature for requests to AWS:

1. For each request parameter, URL-encode the parameter values. URL encoding should be performed as specified in RFC1738, section 2.2. Each parameter should consist of a parameter name, followed by an equal sign (=), followed by the URL-encoded parameter value. There should be no embedded spaces in this string.

   ```
   param-name=encoded-param-value
   ```

2. Form a string by appending each parameter to the end of the string followed by an ampersand (&). The parameter values should be sorted alphabetically by the name of the parameter. There should be no embedded spaces in this string. The Signature parameter should not be included in this string.

   ```
   param-name1=encoded-param-value1&
   param-name2=encoded-param-value2&
   param-name3=encoded-param-value3
   ```

3. Compute an RFC 2104 compliant HMAC, using the Secret AWS Access Key as the "key". This value should be base64 encoded, and then included as the value for the Signature parameter for the request.

Optional parameters not included in the request should not be canonicalized as "empty" parameters. That is, if no value for a ParameterA is specified in the request, there should not be a "ParameterA=" entry in the canonicalized string.

Authenticating SOAP Requests

In cryptography, X.509 is an ITU-T standard for public key infrastructure (PKI). X.509 specifies, amongst other things, standard formats for public key certificates and certification path validation algorithm.

AWS does not implement a full Public Key Infrastructure (PKI). The certificate information is used only to authenticate requests to AWS.

AWS accepts any syntactically and cryptographically valid X.509 certificate. AWS will not verify that certificate is not expired or revoked. Certificates can be self-signed or signed by any key. Certificates generated by AWS are self-signed certificates with no expiration time.

Message Expiration

Signed messages must contain an expiration timestamp, represented as a Expires element within a Timestamp element in the WS-Security header. The Expires element should not have a ValueType attribute, and the Timestamp element may have a Created child element.

Signing a SOAP Request

SoapContext.Security.Elements collections allow adding various WS-Security conformant elements. The following code sample demonstrates how to sign a request:

```
X509SecurityToken crtTkn = new X509SecurityToken(cert);
wse.RequestSoapContext.Security.Tokens.Add(crtTkn);
wse.RequestSoapContext.Security.Elements.Add(new Signature(crtTkn));
```

Authenticating SOAP Requests Using HMAC-SHA1

Every non-anonymous request to an AWS service must contain authentication information to establish the identity of the principal making the request. In SOAP, the authentication information is put into the following elements of the SOAP request:

- AWSAccessKeyId: Your AWS Access Key ID
- Timestamp: This must be a dateTime (http://www.w3.org/TR/xmlschema-2/#dateTime) in the Coordinated Universal Time (Greenwich Mean Time) time zone, such as 2005-01-31T23:59:59.183Z. Authorization will fail if this timestamp is more than 15 minutes away from the clock on AWS servers.
- Signature: The RFC 2104 HMAC-SHA1 digest (http://www.ietf.org/rfc/rfc2104.txt) of the concatenation of "ServiceName" + OPERATION + Timestamp, using your AWS Secret Access Key as the key. For example, in the following sample request, the signature element would contain the HMAC-SHA1 digest of the value "ServiceNameOperationName2005-01-31T23:59:59.183Z"

```
<OperationName xmlns="http://service.amazonaws.com/doc/2006-03-01/">
..< … >
  <AWSAccessKeyId>1D9FVRAYCP1VJS767E02</AWSAccessKeyId>
  <Timestamp>2005-01-31T23:59:59.183Z</Timestamp>
  <Signature>SZf1CHmQ/nrZbsrC13hCZS061yws</Signature>
</OperationName>
```

> **Note**
>
> Due to different interpretations regarding how extra time precision should be dropped, .NET users should take care not to send overly specific time stamps. This can be accomplished by manually constructing DateTime objects with only millisecond precision.

Authenticating SOAP Requests Using X.509 Certificates

Developers can choose to either use their own certificate or use one generated by AWS.

`Using Existing Certificates`. When a developer wants to use his or her own certificate, he or she can upload the certificate (not including the Private Key value) to AWS in order to associate it with the developer's AWS account. The certificate used must include a Base64-encoded Distinguished Encoding Rules (DER) certificate body, and have an accompanying Private Key value. This document does not cover how to obtain or generate certificates other than those provided by AWS.

> **Note**
>
> When a certificate is uploaded, AWS confirms that the certificate is not expired according to the contents of the certificate. AWS does not check certificate revocation list (CRL) to determine if the certificate has been revoked. AWS does not validate the certificate with a certificate authority or any trusted third parties.

If a developer does not already have a certificate, or wants a new certificate for use with AWS, he or she can generate one from the Access Identifiers page of the AWS web site. After logging in to AWS, the Access Identifiers page can be accessed by hovering over *`Your Web Services Account`*, then clicking *`View Access Identifiers`*. The developer can then either create or upload a certificate for use with his or her AWS account. The generated certificate can immediately be downloaded, and is automatically associated with the developer's account. For more information about X.509 certificates, including DER, please see http://en.wikipedia.org/wiki/X.509.

Signing a SOAP Request

SoapContext.Security.Elements collections allow adding various WS-Security conformant elements. The following code sample demonstrates how to sign a request:

```
X509SecurityToken crtTkn = new X509SecurityToken(cert);
wse.RequestSoapContext.Security.Tokens.Add(crtTkn);
wse.RequestSoapContext.Security.Elements.Add(new Signature(crtTkn));
```

Authenticating REST Requests

Every request to SQS must contain authentication information to establish the identity of the principal making the request. In REST, this is done by first putting the headers in a canonical format, then signing the headers using your AWS Secret Access Key.

Signatures with REST Requests

To create a signature to include in a REST request, put your AWS Access Key ID and the signature you computed into the Authorization header:

```
Authorization: AWS AWSAccessKeyId + ":" + \
                   hmac-sha1(VERB + "\n" +
                   CanonicalizedHeaders + "\n" + \
                   CanonicalizedResource) + \
```

Canonicalization for Authorization Header Authentication

When authenticating through the Authorization header, you create the string to be signed by concatenating the request verb with canonicalized headers and the resource that the request is targeting.

The headers used for request signing are:

- Content-MD5
- Content-Type
- date

Headers are canonicalized to CanonicalizedHeaders by:

- Lower-case header name.
- Sorted by header name.
- The values of headers whose names occur more than once should be white space-trimmed and concatenated with comma separators as defined in section 4.2 of RFC 2616.
- Remove any whitespace around the colon in the header.
- Remove any newlines ("\n") in continuation lines.
- Separate headers by new lines ("\n").

Example CanonicalizedHeaders

These headers:

```
B: 2
```

```
A: sqs
```

```
B: 1

C: aws
```

Should result in the following:

```
a:sqs

b:2,1

c:aws
```

CanonicalizedResource is the endpoint URL, with scheme (http: or https:), host, and port number specification removed. For example:

This URL:

```
http://queue.amazonaws.com/myqueue/queuename
```

Becomes this:

```
/myqueue/queuename
```

The string to be signed is formed by appending the REST verb, the content-md5 value, content-type value, date value, and the SQS endpoint value all separated by new lines ('\n'), as follows:

```
Verb + "\n" + Content-MD5 + "\n" + Content-Type + "\n" + Date + "\n" +
queueURL Path
```

The resource is specific to the service to which the request is being made. If the request you are signing is for an ACL, you should include ?acl in the resource part of the canonical string. No other query string parameters should be included, however.

Canonicalization for Query String Authentication

When authenticating via query string parameters, you create the string to be signed by concatenating the request verb with canonicalized headers and the resource that the request is targeting. The headers used for request signing are the same as those for authorization header authentication, except that the Date field is replaced by the Expires parameter. The Expires parameter is the time when you want the signature to expire, specified as the number of seconds since the epoch time.

Thus, the string to be signed is formed by appending the REST verb, content-md5 value, content-type value, expires parameter value, canonicalized x-amz headers (see recipe below), and the resource; all separated by new lines ('\n').

Some important points:

- The string to sign (verb, headers, resource) must be UTF-8 encoded.
- The content-type and content-md5 values are optional, but if you do not include them you must still insert a newline at the point where these values would normally be inserted.
- Some toolkits may insert headers that you do not know about beforehand, such as adding the header 'Content-Type' during a PUT. In most of these cases, the value of the inserted header remains constant, allowing you to discover the missing headers using tools such as Ethereal or tcpmon.
- The value of the Date header must specify a time no more than 15 minutes away from the AWS server's clock.
- The hash function to compute the signature is HMAC-SHA1 defined in RFC 2104 (http://www.ietf.org/rfc/rfc2104.txt), using your Secret Access Key as the key.

Access Control Overview

In SQS, Access Control refers to assigning access rights, or privileges, to an SQS queue or message. Access rights are used to control who has access to perform operations on the queue or message(s) specified. When a queue is created, only the owner of the queue can modify the access control settings on the queue using the ACS actions. The queue owner is determined by the Access Key ID included in the request to create the queue. Ownership cannot be transferred, but it is possible to grant another user `FULLCONTROL` permissions to the queue, meaning that user could perform all actions on the queue, including setting access restrictions.

There are three actions related to Access Control: `AddGrant`, `RemoveGrant`, and `ListGrants`. Each action can be called using any of the supported request formats, which include SOAP, HTTP REST (REST), and HTTP Query (Query).

Note

ACS actions cannot be used with REST requests.

About Users

A user is an individual that can be granted access to a resource. There are three ways that a user can be represented:

```
<Grantee xsi:type="AmazonCustomerByEmail">
  <EmailAddress>chriscustomer@email.com</EmailAddress>
</Grantee>
```

```
<Grantee xsi:type="CanonicalUser">
  <ID>a9a7b886d6fd24a52fe8ca5bef65f89a64e0193f23000e241bf9b1c61be666e9</ID>
  <DisplayName>customerNickname<DisplayName>
</Grantee>
```

- AmazonCustomerByEmail: A user can be identified by the email address that they use to log into their Amazon.com retail web account. The XML format for specifying a grantee by their Amazon customer email address is:
- CanonicalUser: This is the canonical representation of a user as it is returned from a `ListGrants` request. It has an ID and a display name component. The ID is a system assigned string that uniquely identifies the user. The Display name is a a string that identifies the user to a human. The Display Name is calculated for an AWS customer as follows: If the customer has assigned him/herself a public nickname using the Amazon.com retail web site, it is used as the display name. Otherwise, the first component of the customer's email address (the part before the @ symbol) is used. This value is not unique, and may change over time. For example, if a customer changes his or her email address, or updates his or her retail website nickname, the change will be reflected in the Display Name in subsequent ACL responses.

  The XML format for specifying a grantee by their canonical representation is:

About Grantees

A grantee is an abstract type that defines the user identity to which a grant is applied.

About Permissions

A permission, or right, is the type of access that is allowed to a grantee. A grant defines the permissions the grantee has to perform actions on the resource to which the grant is applied. The following grants (permission types) are supported: `FULLCONTROL`, `READ`, and `WRITE`.

Please see ACS Permissions for more information about permission types.

About Grants

A grant is a combination of a grantee and a permission. A grant specifies the permission conferred to the grantee for the queue.

SQS Access Control Permissions

The permission in a grant describes the type of access to be granted to the respective grantee. The following permissions are supported in SQS:

- *`Read:`* When applied to a queue, this grants permission to send and receive the messages in the queue. When applied to an message, this grants permission to read the message body.
- *`Write:`* When applied to a queue, this grants permission to send or delete any message in the queue. This permission is not supported for individual messages within a queue.
- *`FullControl:`* This permission represents the inclusive set of permissions of Read and Write. If FullControl is assigned to another user, that user then has the ability to remove access rights from the owner of the queue. However, only the queue owner can delete a queue.

For more information about how to assign grants, please see the API Reference topics for ACS actions:

- Query Actions
- SOAP Actions
- REST Actions

SQS API Reference

This section of the Developer Guide contains an API reference for the Actions (operations), error codes, and detailed technical information for the Amazon Simple Queue Service API.

Each Action listed in the API Reference contains at least one sample request to help you get started. Use the sample requests as a starting point for developing your own requests. Keep in mind that you should substitute your own Access Key ID (as the value of the AWSAccessKeyId) parameter into the requests before using them. Requests to SQS must also be signed, or authenticated. For more information about authentication, please see the Request Authentication topic.

Simple Queue Service WSDL and Schema Location

The Simple Queue Service (SQS) API is published through a Web Services Description Language (WSDL) and an XML Schema document. The locations of the WSDL and Schema for the Simple Queue Service are listed below.

The latest version of the Simple Queue Service API is 2006-04-01.

Simple Queue Service WSDL Location

The SQS WSDL contains 2 port types: QueueService and MessageQueue, each of which represents and endpoint on which actions can be performed.

http://queue.amazonaws.com/doc/2006-04-01/QueueService.wsdl

Default XML Schema Location

http://queue.amazonaws.com/doc/2006-04-01/QueueService.xsd

Finding the Service Version

The version of the service is defined in the service WSDL, and is simply the date that is embedded within the namespace. In this case, the version is *2006-04-01*. Service versioning ensures that applications that validate against the current or older schemas are not affected when an AWS service adds new elements to the schema. The service version is also required in all access control action requests.

The service version is defined in the namespace of the Web Services Description Language (WSDL) document. The WSDL namespace appears in the first line or element of the WSDL file. The following is the first element from an Simple Queue Service WSDL:

```
<xs:import namespace="http://queue.amazonaws.com/doc/2006-04-01/"
```

The service version also appears in the URL to the WSDL file. Specifically, the URL for the WSDL for this version of SQS is:

http://queue.amazonaws.com/doc/2006-04-01/QueueService.wsdl

Retrieving an XML Schema (XSD)

The XML schema responses for requests is also versioned. Just as you can access specific WSDL versions by including the version number (or date) in the WSDL URL, you can also access schemas by inserting the version number in the schema URL. For this version of SQS, the URL to access the schema associated with the QueueService.wsdl is:

http://queue.amazonaws.com/doc/2006-04-01/QueueService.xsd

Accessing a Specific Service Version

For all requests, you must explicitly request the version you want to use. Specifying the version parameter ensures that the service does not return response elements that your application is not designed to handle. In REST and Query requests, this is done by including the *`Version`* parameter in your request. For SOAP, this is accomplished by the namespace of the first child element of the `Body>` element.

Note

When accessing the WSDL or Schema by opening the URL in a Web browser, you must view the source of the loading page to view the actual .wsdl or .xsd rather than the browser's interpretation of them.

Response Groups

Response Messages, SOAP and Query

In response an action request, SQS returns an XML data structure that contains the results of the request. This data conforms to the SQS schema.

Other than the use of a message envelope in the case of SOAP, the schema for the results is the same for both SOAP and Query responses. The SOAP WSDL imports an XSD file to define the response messages, and Query and REST users can access the XSD file directly. For more information, see WSDL and Schema Locations.

The Structure of a Response

The response message is returned in an XML element named after the action. For example, the `CreateQueue` action returns a response element named *`CreateQueueResponse`*. This element contains child elements associated with the action performed.

Each response also contains information about whether the request succeeded or failed. When a request is successful, a *`ResponseStatus`* element is returned containing a *`StatusCode`* element with a value of "Success."

If a request is unsuccessful, no *`ReturnStatus`* element is included in the request. Instead, the response contains an *`Errors`* element, with one or more *`Error`* elements. Each *`Error`* includes:

- a *`Code`* that identifies the type of error the occurred.
- a *`Message`* that describes the error condition in a human-readable form.

SQS Query Reference

HTTP Query-based requests are defined as any HTTP requests using the HTTP verb `GET` or `POST` and either a Query parameter named Action or Operation. Action is used throughout this documentation. If an HTTP request is sent that does not contain either of these parameters, it is considered a REST request. Please see REST Actions for more information.

The Simple Queue Service Query API implements the following actions:

- Common Query Parameters
- CreateQueue
- ListQueues
- DeleteQueue
- SendMessage
- ReceiveMessage
- PeekMessage
- DeleteMessage
- SetVisibilityTimeout
- GetVisibilityTimeout
- AddGrant
- ListGrants
- RemoveGrant

Common Query Parameters

The following parameters must be included in each Query request to SQS:

Parameter Name	Description	Example Value
Action	Indicates the action to perform.	`CreateQueue`
Version	Required. The API version to use, as specified in the WSDL.	2006-04-01
AWSAccessKeyId	The Access Key ID for the request sender. This identifies the account	0AS7253JW73RRM652K02

Parameter Name	Description	Example Value
	which will be charged for usage of the service. The account with which the Access Key ID is associated must be signed up for SQS, or requests will not be accepted..	
`Timestamp`	The date and time at which the request is signed, in the format YYYY-MM-DDThh:mm:ssZ, as specified in the ISO 8601 standard.	2006-07-07T15:04:56Z
`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.	2006-07-07T15:04:56Z
`Signature`	A request signature is calculated by concatenating the values of each request parameter included in the request, in alphabetical order, and then calculating an RFC 2104-compliant HMAC-SHA1 hash, using the Secret AWS Access Key as the key. Please see Request Authentication for more information about calculating signatures.	Qnpl4Qk/7tINHzfXCiT7VbBatDA=

Note

The *Timestamp* parameter can be used instead of *Expires*. Requests must include either *Timestamp* or *Expires*, but cannot contain both.

When passing these values from a web browser, either via the address line or navigation bar, they must be URL-encoded. This is true for any Query parameter passed to SQS using a browser, and is typically necessary in the *Signature*, *MessageId*, and *MessageBody* parameters.

Return Elements

The following common elements are also returned within a debug block:

- RequestId: an ID that uniquely identifies this request.
- Status: String, one of Success, Fail, Warnings, or Errors.
- Warnings: Array of (String warning code, String message) if there were any warnings.
- Errors: Array of (String error code, String message) if there were any errors.

CreateQueue (Query)

The CreateQueue action creates a new queue. An optional queue name may be provided to associate with the queue for future reference, but the queue name must be unique among the queues associated with the Access Key ID provided in the request. All queues are assigned a queue ID, which is unique to each AWS account signed up for SQS. The URL to the queue, the Queue URL, is used to specify the queue on which to perform actions.

If no value for *QueueName* is provided, a unique string is generated for the queue name.

If the queue already exists, CreateQueue returns the Queue URL with a error indicating that the queue already exists.

When a queue is created, Full Control access rights are granted to the AWS account associated with the Access Key ID included in the request. This identifies the "owner" of the queue. Initially, only the owner of the queue can grant or deny access rights to the queue and messages in it.

Note

The AWS account associated with the Access Key ID must be signed up for Simple Queue Service for the request to succeed, as Simple Queue Service is not a free service.

The default value for *VisibilityTimeout* is also set when a queue is created. Visibility assists in assuring that messages are only delivered once. *VisibilityTimeout* refers to the length of time, in seconds, that a message will not be returned to a request to retrieve it after already being returned to a similar request. In other words, if a message is retrieved from a queue, it will not be retrieved again from another request for messages in that queue until the druation of *VisibilityTimeout* has passed. This value applies to all messages in the queue, unless otherwise specified by another action on the message.

Validation

CreateQueue requests are validated on the following:

- The value specified for *QueueName* must be alphanumeric of length 1 to 80.

Returns

Returns Success and a Queue URL, or an error response if unsuccessful.

Request Parameters

Name	Description	Type	Value
`QueueName`	The name to use for the Queue created. The Queue name must be unique for all queues created by the given Access Key ID.	Optional	An alphanumeric string specifying the name of the Queue to create. Maximum 80 characters.
`DefaultVisibilityTimeout`	Sets the default visibility timeout for this Queue. If this parameter is not included, the default value is set to 30 seconds.For more information, please see SetVisibilityTimeout.	Optional	Integer

Example CreateQueue Request (Query)

The following example of an `CreateQueue` action creates a new Queue named queue2.

Sample Request

```
http://queue.amazonaws.com/
?Action=CreateQueue
&QueueName=queue2
&AWSAccessKeyId=0GS7573JW74RZM612K0A
&Version=2006-04-01
&Expires=2007-01-12T12:00:00Z
&Signature=Dqlp3Sd6ljTUA9Uf6SGtEExwUQE=
```

Sample Response

```
<CreateQueueResponse
  xmlns=http://queue.amazonaws.com/doc/2006-04-01/
  xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
  xsi:type=CreateQueueResponse>
    <QueueUrl>
        http://queue.amazonaws.com/A23E9WXPHGOG29/queue2
    <QueueUrl>
    <ResponseStatus>
      <StatusCode>Success</StatusCode>
      <RequestId>cb919c0a-9bce-4afe-9b48-9bdf2412bb67</RequestId>
    </ResponseStatus>
</CreateQueueResponse>
```

Error Responses

Error	Description
InvalidParameterValue	QueueName did not pass validation.

Example Error Response

Specifying a QueueName that is too long or not alphanumeric results in the following error:

```
<Response>
    <Errors>
        <Error>
            <Code>
                InvalidParameterValue
            </Code>
            <Message>
              Value (quename_nonalpha) for parameter QueueName is invalid.
              Must be an alphanumeric String of 1 to 80 in length
            </Message>
        </Error>
    </Errors>
    <RequestID>
        42d59b56-7407-4c4a-be0f-4c88daeea257
    </RequestID>
</Response>
```

ListQueues (Query)

The `ListQueues` action returns a list of the queues associated with the AWS account represented by the Access Key ID included in the request. A maximum of 10,000 queue URLs are returned. If a value is specified for the optional *`QueueNamePrefix`* parameter, only those queues with a queue name beginning with the value specified are returned. The queue name is specified in the *`QueueName`* parameter when a queue is created.

Validation

`ListQueues` requests do not contain any parameter values that require validation.

Returns

The `ListQueues` action returns a list of queue URLs representing the queues associated with the AWS account for the Access Key ID included in the request.

Request Parameters

The following table lists the parameters of the operation.

Name	Description	Type	Value
`QueueNamePrefix`	Optional. This parameter can be used to filter results returned. When specified, only queues with queue names be-	String	User-defined alphanumeric String

Name	Description	Type	Value
	ginning with the specified string are returned.		

Example ListQueues Request (Query)

The following example `ListQueues` request returns the queues owned by the Access Key ID in the request that have a queue name that begins with "T".

Sample Request

```
http://queue.amazonaws.com/
?Action=ListQueues
&QueueNamePrefix=T
&AWSAccessKeyId=0AS7553JW73RRM642K02
&Version=2006-04-01
&Expires=2007-01-12T12:00:00Z
&Signature=Qnpl4Qk/7tINHzfXCiT7VbBatDA=
```

Sample Response

The `ListQueues` action returns the queues associated with the specified Access Key ID with names that begin with "T".

```
<ListQueuesResponse
  xmlns=http://queue.amazonaws.com/doc/2006-04-01/
  xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
  xsi:type=ListQueuesResponse>
    <Queues>
        <QueueUrl>
            http://queue.amazonaws.com/A29E9VSPHGOG23/Toast
        </QueueUrl>
        <QueueUrl>
            http://queue.amazonaws.com/A29E9VSPHGOG23/Test
        </QueueUrl>
    </Queues>
    <ResponseStatus>
      <StatusCode>Success</StatusCode>
      <RequestId>cb919c0a-9bce-4afe-9b48-9bdf2412bb67</RequestId>
    </ResponseStatus>
</ListQueuesResponse>
```

DeleteQueue (Query)

Deletes the queue specified by the queue URL. A Queue is deleted only if it is does not contain any messages. Use DeleteMessage to delete messages from the queue.

Validation

`DeleteQueue` requests are validated on the following:

- The queue specified must exist.

Returns

Returns Success if the request is successful, or an error response if unsuccessful.

Request Parameters

The `DeleteQueue` action takes no input parameters other than those common to all requests to SQS.

Example DeleteQueue Request (Query)

The following example of a `DeleteQueue` request deletes the queue specified by the provided Queue URL.

Sample Request

```
http://queue.amazonaws.com/QueueId/queue1
?Action=DeleteQueue
&AWSAccessKeyId=0GS7553JW74RRM612K02
&Version=2006-04-01
&Expires=2007-01-12T12:00:00Z
&Signature=CN2SbNq%2B2Vw1W3lbc7wpM5gzDHE=
```

Sample Response

```
<DeleteQueueResponse>
    <ResponseStatus>
      <StatusCode>Success</StatusCode>
      <RequestId>cb919c0a-9bce-4afe-9b48-9bdf2412bb67</RequestId>
    </ResponseStatus>
</DeleteQueueResponse>
```

SendMessage (Query)

The `SendMessage` action delivers a message to the specified queue. A queue is represented by a queue URL. The content of the message is specified in the *`MessageBody`* parameter, and can be any text. However, the message body text must be URL-encoded. The total string length of *`MessageBody`* cannot exceed 256K.

For more information about URL encoding, please see http://en.wikipedia.org/wiki/URL_encoding.

Validation

`SendMessage` requests are validated on the following:

- *`MessageBody`* must exist and be a string of length 1 byte to 256K.
- The Queue specified must exist. The URL to the queue must have been returned by a `CreateQueue` or `ListQueues` request.

Returns

This action, if successful, returns Success and a string representing the ID of the message sent, which is used as the value of the *MessageId* parameter when performing actions on that message. This action returns an error response if unsuccessful.

Request Parameters

Name	Description	Type	Value
MessageBody	Required. A String representing the body of the message to send. The total string length of the message body cannot exceed 256K.	String	The content of the message. The message can contain any valid string character, but must be URL-encoded so that the string does not contain any characters that would not be valid in an HTTP URL.

Example SendMessage Request (Query)

The following example SendMessage request sends a message containing "Your Message Text" as the body of the message to the specified queue.

Sample Request

```
http://queue.amazonaws.com/A29E9VSPHGOG23/queue1
?Action=SendMessage
&MessageBody=Your%20Message%20Text
&AWSAccessKeyId=0GS7553JW74RRM612K02
&Version=2006-04-01
&Expires=2007-01-12T12:00:00Z
&Signature=lB/P67vCvGlDMBQ1dofZxg8E8SU=
```

Sample Response

```
<SendMessageResponse>
    <MessageId>
        1EDR8H6XFYE9PGV3FGQQ|04WQYHSQ39E6Y1K6EDFS|0QE42ST4KW7RK9HSY074
    </MessageId>
    <ResponseStatus>
      <StatusCode>Success</StatusCode>
      <RequestId>cb919c0a-9bce-4afe-9b48-9bdf2412bb67</RequestId>
    </ResponseStatus>
</SendMessageResponse>
```

Error Response

Error	Description
MissingParameter	No value for *MessageBody* was supplied.
InvalidURI	The URL for the queue is not valid, or was not an URL returned by a CreateQueue or ListQueues request.

ReceiveMessage (Query)

Retrieves one or more messages from the queue specified, returning the message body and message ID of each message. Messages returned by this action stay in the queue until deleted. However, once a message is returned to a ReceiveMessage request, it will not be returned on subsequent ReceiveMessage requests until the duration of the *VisibilityTimeout* has passed. Please see SetVisibilityTimeout for more information.

Note

To view a message without locking it, in other words, without affecting the visibility state, use PeekMessage.

Validation

ReceiveMessage action requests are validated on the following:

- The Queue specified must exist. The URL to the queue must have been returned by a CreateQueue or ListQueues request.
- VisibilityTimeout, if used, must be an integer between 0 and 86400.
- *NumberOfMessages*, if used, must be an integer between 1 and 255.

Returns

This action returns Success, the message ID and message body of each available (those not restricted by the visibility timeout setting) message if successful. It returns an error response if unsuccessful.

Request Parameters

Name	Description	Type	Value
NumberOfMessages	Optional. Specifies the maximum number of messages to return. If the number of messages in the queue is less than value specified by *NumberOfMessages*, then the number of messages returned is up to the number of messages in the queue. Not necessarily all the messages in the queue will be returned. If no value is provided, the default	Integer	A value between 1 and 256.

Name	Description	Type	Value
	value of 1 is used.		
`VisibilityTimeout`	Optional. The duration, in seconds, that the messages are visible in the queue. If no value is specified, the default value for the queue is used. The default value is set using `CreateQueue` or `SetVisibiltityTimeout`.	Integer	0 to 86400 seconds (maximum 24 hours)

Example ReceiveMessage Request (Query)

Sample Request

```
http://queue.amazonaws.com/A29E9VSPHGOG23/queue1
?Action=ReceiveMessage
&AWSAccessKeyId=0GS7553JW74RRM612K02
&Version=2006-04-01
&Expires=2007-01-12T12:00:00Z
&Signature=CN2SbNq%2B2Vw1W3lbc7wpM5gzDHE=
```

Sample Response

```
<ReceiveMessageResponse>
    <Messages>
        <MessageId>
           17VXQHSGX0SG4ZEPPK7R|0QE42ST4KW7RK9HSY074|0Z4AN912X0H2EP8BV6XJ
        </MessageId>
        <MessageBody>foo</MessageBody>
    </Messages>
    <ResponseStatus>
      <StatusCode>Success</StatusCode>
      <RequestId>cb919c0a-9bce-4afe-9b48-9bdf2412bb67</RequestId>
    </ResponseStatus>
</ReceiveMessageResponse>
```

Error Responses

Error	Description
`InvalidParameterValue`	The value specified for *`VisibilityTimeout`* or *`NumberOfMessages`* is out of range.
InvalidURI	The URL for the queue is not valid, or is not an URL returned by a `CreateQueue` or `ListQueues` request.

DeleteMessage (Query)

The `DeleteMessage` action removes the specified message from the queue. Messages stay in the queue until they are deleted with a `DeleteMessage` request.

Validation

`DeleteMessage` requests are validated on the following:

- The message specified must exist.

Note

Even if the message is locked by another reader due to the visibility timeout setting it will still be deleted from the queue.

Returns

Success and the message ID if successful, or an error code if unsuccessful.

Request Parameters

Name	Description	Type	Value
`MessageId`	The ID of the message to delete.	String	A message ID value returned from a SendMessage request.

Example DeleteMessage Request (Query)

The following example `DeleteMessage` request demonstrates how to delete a message from the queue by specifying the message ID of the messasge.

Sample Request

```
http://queue.amazonaws.com/A29E9VSPHGOG23/queue1
?Action=DeleteMessage
&AWSAccessKeyId=0GS7553JW74RRM612K02
&Version=2006-04-01
&MessageId=17VXQHSGX0SG4ZEPPK7R%7C0QE42ST4KW7RK9HSY074%7C0Z4AN912X0H2EP8BV6XJ
&Expires=2007-01-12T12:00:00Z
&Signature=CN2SbNq%2B2Vw1W3lbc7wpM5gzDHE=
```

Sample Response

```
<DeleteMessageResponse>
    <Message>
            <MessageId>
            17VXQHSGX0SG4ZEPPK7R|0QE42ST4KW7RK9HSY074|0Z4AN912X0H2EP8BV6XJ
            </MessageId>
    </Message>
    <ResponseStatus>
```

```
      <StatusCode>Success</StatusCode>
      <RequestId>cb919c0a-9bce-4afe-9b48-9bdf2412bb67</RequestId>
    </ResponseStatus>
</DeleteMessageResponse>
```

Error Response

Error	Description
MissingParameter	*MessageId* parameter was not present in the request.
InvalidURI	The URL for the queue is not valid, or was not an URL returned by a CreateQueue or ListQueues request.
InvalidParameterValue	No message exists with the ID specified.

PeekMessage (Query)

The PeekMessage action returns a preview of the message specified in the *MessageId* parameter. The message is returned regardless of the *VisibilityTimeout* state on the queue. The visibility state is not modified when PeekMessage is used, thereby not affecting which messages get returned from a subsequent ReceiveMessage request.

Validation

A PeekMessage request is validated on the following:

- The Queue specified must exist. The URL to the queue must have been returned by a CreateQueue or ListQueues request.
- MessageId must exist and be a value returned by a ReceiveMessage or SendMessage request.

Returns

This action returns Success and the message ID of the message specified if successful, or an error response if unsuccessful.

Request Parameters

Name	Description	Type	Value
MessageId	Required. A list of Message IDs representing the messages to return.	String	17VXQHSGX0SG4ZEPPK7R|0QE42ST4KW7RK9HSY074|0Z4AN912X0H2EP8BV6XJ

Example PeekMessage Request (Query)

The following example PeekMessage request returns the message associated with the message ID specified.

Sample Request

```
http://queue.amazonaws.com/A29E9VSPHGOG23/queue1
?Action=PeekMessage
&AWSAccessKeyId=0GS7553JW74RRM612K02
&Version=2006-04-01
&MessageId=17VXQHSGX0SG4ZEPPK7R%7C0QE42ST4KW7RK9HSY074%7C0Z4AN912X0H2EP8BV6XJ
&Expires=2007-01-12T12:00:00Z
&Signature=CN2SbNq%2B2Vw1W3lbc7wpM5gzDHE=
```

Sample Response

```
<PeekMessageResponse>
    <Message>
        <MessageId>
        17VXQHSGX0SG4ZEPPK7R|0QE42ST4KW7RK9HSY074|0Z4AN912X0H2EP8BV6XJ
        </MessageId>
        <MessageBody>foo</MessageBody>
    </Message>
    <ResponseStatus>
      <StatusCode>Success</StatusCode>
      <RequestId>cb919c0a-9bce-4afe-9b48-9bdf2412bb67</RequestId>
    </ResponseStatus>
</PeekMessageResponse>
```

Error Response

Error	Description
MissingParameter	*MessageId* parameter was not present in the request.
InvalidURI	The URL for the queue is not valid, or was not an URL returned by a CreateQueue or ListQueues request.
InvalidParameterValue	No message exists with the ID specified.

SetVisibilityTimeout (Query)

Sets the amount of time, *VisibilityTimeout*, messages are hidden from subsequent read requests after being retrieved by a ReceiveMessage request. The value is set in seconds, and measured from the time of the request.

To set the *VisibilityTimeout* on a select messages in the queue, specify the ID for that message using the *MessageId* parameter. If no message ID is provided, the *VisibilityTimeout* value is set for all messages in the queue.

The visibility timeout for a message indicates whether to return the message to a request. If a message is returned, it will not be returned on subsequent ReceiveMessage requests for the duration of the *VisibilityTimeout* period. Once that duration has passed, the message will again be returned to a ReceiveMessage request unless it is deleted from the queue.

Validation

The following criteria are validated in a `SetVisibilityTimeout` request.

- *VisibilityTimeout* must be an integer between 0 and 86400.
- The specified resource must exist (the URL to the resource must have been returned by an SQS request).

Returns

This action returns Success if successful, or an error code if unsuccessful.

Request Parameters

Name	Description	Type	Value
VisibilityTimeout	Required. The duration, in seconds, that the message specified by *MessageIds* is visible in the queue.	Integer	0 to 86400 seconds (maximum 24 hours)
MessageIds	Optional. Specifies a message ID for which to set the *VisibilityTimeout* value.	String	

Example SetVisibilityTimeout Request (Query)

The following example sets the visibility timeout to 35 seconds for all messages in the specified queue.

Sample Request

```
http://queue.amazon.com/A29E9VSPHGOG23/queue2
?Action=SetVisibilityTimeout
&VisibilityTimeout=35
&AWSAccessKeyId=0GS7553JW74RRM612K02
&Expires=2007-01-12T12:00:00Z
&Signature=Dqlp3Sd6ljTUA9Uf6SGtEExwUQE=
```

Sample Response

```
<SetVisibilityTimeoutResponse>
     <ResponseStatus>
      <StatusCode>Success</StatusCode>
      <RequestId>cb919c0a-9bce-4afe-9b48-9bdf2412bb67</RequestId>
    </ResponseStatus>
</SetVisibilityTimeoutResponse>
```

Error Responses

Error	Description
InvalidParameterValue	*VisibilityTimeout* was not an integer between 0 and 86400.
InvalidURI	The URL for the queue is not valid, or was not an URL returned by a CreateQueue or ListQueues request.

GetVisibilityTimeout (Query)

The GetVisibilityTimeout action gets the *VisibilityTimeout* value set on the queue specified.

The *VisibilityTimeout* is the amount of time a message in a queue is not returned to ReceiveMessage requests. For example, message A is in a queue and has a visibility timeout set at 30 seconds. A ReceiveMessage request is made on that queue, and Message A is returned. A subsequent ReceiveMessage reqeust is made in 10 seconds (within duration of *VisibilityTimeout* of 30 seconds). Message A is not returned to that request. Another request is made 20 seconds after the first request. Message A is again not returned. Any request made after 30 seconds has passed will return Message A until it is deleted from the queue.

Messages that are meant to be received only once should be deleted, using DeleteMessage within the duration of the *VisibilityTimeout*.

Validation

CreateQueue requests are validated on the following:

- The value specified for *QueueName* must be alphanumeric of length 1 to 20.

Returns

Returns Success and the current visibility timeout setting if successful, or an error response if unsuccessful.

Request Parameters

This operation has no input parameters other than those common to all requests to SQS. Please see Query Parameters for more information.

Example GetVisibilityTimeout Request (Query)

The following example GetVisibilityTimeout request returns the *VisibilityTimeout* value for the queue specified.

Sample Request

```
http://queue.amazonaws.com/A29E9VSPHGOG23/queue2
?Action=GetVisibilityTimeout
&AWSAccessKeyId=0GS7553JW74RRM612K02
&Version=2006-04-01
&Expires=2007-01-12T12:00:00Z
```

```
&Signature=Dqlp3Sd6ljTUA9Uf6SGtEExwUQE=
```

Sample Response

```
<GetVisibilityTimeoutResponse>
    <VisibilityTimeout>
      35
    </VisibilityTimeout>
    <ResponseStatus>
      <StatusCode>Success</StatusCode>
      <RequestId>cb919c0a-9bce-4afe-9b48-9bdf2412bb67</RequestId>
    </ResponseStatus>
</GetVisibilityTimeoutResponse>
```

Error Responses

Error	Description
InvalidURI	The URL for the queue is not valid, or was not an URL returned by a `CreateQueue` or `ListQueues` request.

AddGrant (Query)

The `AddGrant` operation gives the specified user permission to access the queue specified. The `AddGrant` operation adds a grant to the specified resource, which adds permissions for the grantee to that queue.

Validation

`ListGrants` requests are validated on the following:

- The specified queue must exist.

Returns

The AddGrant request returns Success if the action succeeded, or an error response if unsuccessful.

Request Parameters

Name	Description	Type	Value
Action	Specifies the action (or operation) to perform.	Required. String.	`AddGrant`
Version	The API version to use, as specified in the WSDL.	Required. Date.	2006-04-01
AWSAccessKeyId	Your Access Key ID	Required. The AWS generated Access Key ID associated with the	

Name	Description	Type	Value
		owner of the queue.	
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.	String	2006-07-07T15:04:56Z
Signature	A request signature is calculated by concatenating the values of each request parameter included in the request, in alphabetical order, and then calculating an RFC 2104-compliant HMAC-SHA1 hash, using the Secret AWS Access Key as the key. Please see the Request Authentication topic for more information about calculating signatures.	HMAC-SHA1 digest calculated from the request parameters.	Qn-pl4Qk/7tINHzfXCiT7VbBatDA=
QueueName	Specifies the queue to which the grant is applied.	Required. The name of the queue, as specified when the queue was created.	MyQueue
Grantee.EmailAddress	The email address of the user for which the grant is added.	Required. String.	email_alias@server_domain.com
Permission	The grant, or permission, to add for the Grantee. Please see ACS Permissions for a list of available grants.	String.	Read

Sample AddGrant Request (Query)

The following sample demonstrates assigning Read permissions to the specified queue.

Sample Request

```
http://queue.amazonaws.com/A29E9VSPHGOG23/queue2
?Action=AddGrant
&Version=2006-04-01
&AWSAccessKeyId={ACCESS_KEY_ID]
&Expires=[TIME]
```

```
&Signature=[SIGNATURE]
&Grantee.EmailAddress=[EMAIL_ADDRESS]
&Permission=Read
```

Sample Response

```
<AddGrantResponse xmlns=http://access.amazonaws.com/doc/2006-01-01/>
    <ResponseStatus>
        <StatusCode>Success</StatusCode>
    </ResponseStatus>
</AddGrantResponse>
```

ListGrants (Query)

The `ListGrants` operation lists the grants (permissions) for the use of this queue. Only the owner of the queue can use the ListGrants action.

Any user calling any Access Control actions must have FullControl access to the specified queue for these operations to succeed. If the sender of the request does not have FullControl the operations will fail with an "Access Denied" error.

Returns

This action, if successful, returns Success and a list of grants assigned to the specified user. If unsuccessful, it returns an error response.

Request Parameters

Name	Description	Type	Value
Action	The action to perform.	String	ListGrants
Version	The API version to use, as specified in the WSDL.	Required. Date.	2006-04-01
AWSAccessKeyId	The Acccess Key ID of the owner of the queue.	Required. String	Your Access Key ID.
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.	String	2006-07-07T15:04:56Z
Signature	A request signature is calculated by concatenating the values of each request parameter included in the request, in alphabetical order, and then calculating an	HMAC-SHA1 digest calculated from the request parameters.	Qn-pl4Qk/7tINHzfXCiT7VbBatDA=

Name	Description	Type	Value
	RFC 2104-compliant HMAC-SHA1 hash, using the Secret AWS Access Key as the key. Please see the Request Authentication topic for more information about calculating signatures.		
QueueName	Specifies the queue to which the grant is applied.	Required. The name of the queue, as specified when the queue was created.	MyQueue
Grantee.EmailAddress	The email address of the entity for which access rights are returned.	String	email_alias@server_domain.com
Permission	If specified, the type of grant, or permission, to list for the specified Grantee. Please see ACS Permissions for a list of available grants.	Optional. String.	Read

Sample ListGrant Request (Query)

This ListGrants Query request lists all the Grants with Read permission for the user with the specfied email address

```
http://queue.amazonaws.com/A29E9VSPHGOG23/queue2
?Action=ListGrants
&Version=2006-04-01
&AWSAccessKeyId={ACCESS_KEY_ID]
&Expires=[TIME]
&Signature=[SIGNATURE]
&Grantee.EmailAddress=[EMAIL_ADDRESS]
&Permission=Read
```

If the Permission is omitted, all the grants for a user will be listed. This ListGrants Query request lists all the grants for the user identified by the specified Canonical User ID.

```
http://queue.amazonaws.com/A29E9VSPHGOG23/queue2
?Action=ListGrants
&Version=[VERSION]
&AWSAccessKeyId={ACCESS_KEY_ID]
&Expires=[TIME]
&Signature=[SIGNATURE]
&Grantee.CanonicalUser=93ddb081b6ca9f35fbf3626dde06b73854d43b9a7b6305201550c9
```

```
0aa7bd1eef
```

If both the Permission and the Grantee are omitted then all the grants for all the users will be listed. This ListGrants query request lists all the grants on this resource.

```
http://service.amazonaws.com/
?Action=ListGrants
&Version=[VERSION]
&AWSAccessKeyId={ACCESS_KEY_ID]
&Expires=[TIME]
&Signature=[SIGNATURE]
```

Sample Response

```
<ListGrantsResponse xmlns="http://access.amazonaws.com/doc/2006-01-01/">
  <GrantList>
    <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="CanonicalUser">
<ID>79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be>/ID>
      <DisplayName>filesinc>/DisplayName>
    </Grantee>
    <Permission>ReceiveMessage>/Permission>
  </GrantList>
  <GrantList>
    <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="CanonicalUser">
<ID>a9a7b886d6fd24a52fe8ca5bef65f89a64e0193f23000e241bf9b1c61be666e9>/ID>
      <DisplayName>chriscustomer>/DisplayName>
    </Grantee>
    <Permission>FullControl>/Permission>
  </GrantList>
  <ResponseStatus>
    <StatusCode>Success>/StatusCode>
  </ResponseStatus>
</ListGrantsResponse>
```

RemoveGrant (Query)

The `RemoveGrant` action revokes a grant (permission) for the use of this queue. Only the owner of the queue, or grantee with Full Control permissions, can add or remove grants.

Validation

`AddGrant` requests are validated on the following:

- The specified queue must exist.

Returns

Returns Success if successful, or an error response if unsuccessful.

Request Parameters

The following table lists the parameters for the RemoveGrant operation.

Name	Description	Type	Value
Action	Specifies the action (or operation) to perform.	Required. String.	RemoveGrant
Version	The API version to use, as specified in the WSDL.	Required. Date.	2006-04-01
AWSAccessKeyId	Your Access Key ID	Required. The AWS generated Access Key ID associated with the owner of the queue.	
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.	String	2006-07-07T15:04:56Z
Signature	A request signature is calculated by concatenating the values of each request parameter included in the request, in alphabetical order, and then calculating an RFC 2104-compliant HMAC-SHA1 hash, using the Secret AWS Access Key as the key. Please see the Request Authentication topic for more information about calculating signatures.	HMAC-SHA1 digest calculated from the request parameters.	Qn-pl4Qk/7tINHzfXCiT7VbBatDA=
QueueName	Specifies the queue to which the grant is applied.	Required. The name of the queue, as specified when the queue was created.	MyQueue
Grantee.EmailAddress	The email address of the user for which the grant is removed.	Required. String.	email_alias@server_domain.com
Permission	The grant, or permission, to add for the Grantee. Please see ACS Permissions for a list of available grants.	String.	Read

Example RemoveGrant Request (Query)

The following example `RemoveGrant` request removes all access rights for the email address specified to "queue2".

Sample Request

```
http://queue.amazonaws.com/A29E9VSPHGOG23/queue2
?Action=RemoveGrant
&Version=2006-04-01
&AWSAccessKeyId={ACCESS_KEY_ID]
&Expires=[TIME]
&Signature=[SIGNATURE]
&Grantee.EmailAddress=[EMAIL_ADDRESS]
&Permission=Read
```

Common Query Errors

The following list includes the common error codes that can be returned from Query requests. Errors specific to an action are listed in the topic about that action.

Error	Reason	Sample Response

MissingParameter	A required parameter is missing	`<Response>`
		`    <Errors>`
		`        <Error>`
		`            <Code>MissingParameter</Code>`
		`            <Message>`
		`                The request has a missing`
		`parameter Action`
		`            </Message>`
		`        </Error>`
		`    </Errors>`
		`    <awsRequestId>`
		`        427cce5e-950b-4f82-b30d-d232796a81b1`
		`    </awsRequestId>`
		`    <MissingParameterName>`
		`        Action`
		`    </MissingParameterName>`
		`</Response>`

AccessFailure	A value used for authentication could not be validated, such as *`Signature`*.	`<Response>`
		`    <Errors>`
		`        <Error>`
		`            <Code>AccessFailure</Code>`
		`            <Message>`
		`                AWS was not able to grant ac-`
		`cess to the QueueService service`
		`            </Message>`
		`        </Error>`
		`    </Errors>`
		`    <awsRequestId>`
		`        24f372aa-f7bc-41e4-ad7b-5ec24be5b2f1`
		`    </awsRequestId>`
		`    <ServiceName>QueueService</ServiceName>`
		`</Response>`

Error	Reason	Sample Response
InvalidParameterCombination	Two parameters were specified that cannot be used together, such as *`Timestamp`* and *`Expires`*	
ServiceUnavailable	Some required server needed by SQS is unavailable. This error is often temporary and the request be resent after a short wait.	
InternalError	There is an internal problem with SQS which cannot be resolved by the sender.	

SQS SOAP Reference

The Simple Queue Service SOAP API implements the following actions:

- CreateQueue
- ListQueues
- DeleteQueue
- SendMessage
- ReceiveMessage
- DeleteMessage
- PeekMessage
- SetVisibilityTimeout
- GetVisibilityTimeout
- AddGrant
- ListGrants
- RemoveGrant

Common Return Elements

The following elements are also returned in the Response Status block:

- RequestId: an ID that uniquely identifies this request
- Status: String, one of Success, Warnings, or Errors
- Warnings: Array of (String code, String message) if there were any warnings
- Errors: Array of (String code, String message) if there were any errors
- For SOAP, in the case of an error, the action will return a SOAP fault that contains the code and message.

Common SOAP Parameters

The following parameters must be included in each SOAP request to SQS:

Parameter Name	Description	Example Value
Action	Indicates the action to perform on the specified queue.	`CreateQueue`

Parameter Name	Description	Example Value
AWSAccessKeyId	The Access Key ID for the request sender. This identifies the account which will be charged for usage of the service. The account with which the Access Key ID is associated must be signed up for SQS, or requests will not be accepted..	0AS7253JW73RRM652K02
Timestamp	The date and time at which the request was signed, in the format YYYY-MM-DDThh:mm:ssZ, as specified in the ISO 8601 standard. **Note** The *`Expires`* parameter can be used instead of *`Timestamp`*. Requests must include either *`Timestamp`* or	2006-07-07T15:04:56Z

Parameter Name	Description	Example Value
	, but cannot contain both.	
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. **Note** The `Timestamp` parameter can be used instead of	2006-07-07T15:04:56Z

Parameter Name	Description	Example Value
	. Requests must include either `Timestamp` or	

Parameter Name	Description	Example Value
	, but cannot contain both.	
Signature	A request signature is calculated by concatenating the values of each request parameter included in the request, in alphabetical order, and then calculating an RFC 2104-compliant HMAC-SHA1 hash, using the Secret AWS Access Key as the key. Please see Request Authentication for more information about calculating signatures.	Qnpl4Qk/7tINHzfXCiT7VbBatDA=

CreateQueue (SOAP)

The `CreateQueue` action creates a new queue. An optional queue name may be provided to associate with the queue for future reference, but the queue name must be unique among the queues associated with the Access Key ID provided in the request. All queues are assigned a queue ID, which is unique to each AWS account signed up for SQS. The URL to te queue, the queueURL, is used to specify the queue on which to perform actions.

If no value for *`QueueName`* is provided, a unique string is generated for the queue name.

If the queue already exists, `CreateQueue` returns the queue's URL with a error indicating that the queue already exists.

When a queue is created, Full Control access rights are granted to the AWS account associated with the Access Key ID value included in the request. This identifies the "owner" of the queue. Initially, only the owner of the queue can grant or deny access rights to the queue and messages in it.

> **Note**
>
> The AWS account associated with the Access Key ID must be signed up for Simple Queue Service for the request to succeed, as Simple Queue Service is not a free service.

The default value for *`VisibilityTimeout`* is also set when a queue is created. Visibility assists in assuring that messages are only delivered once. *`VisibilityTimeout`* refers to the length of time, in seconds, that a message will not be returned to a request to retrieve it after already being returned to a similar request. In other words, if a message is retrieved from a queue, it will not be retrieved again from another request for messages in that queue until the druation of *`VisibilityTimeout`* has passed. This value applies to all messages in the queue, unless otherwise specified by another action on the message.

Validation

`CreateQueue` requests are validated on the following:

- The value specified for *`QueueName`* must be alphanumeric of length 1 to 20.

Returns

Returns a Queue, or Fail and an error response if unsuccessful.

Request Parameters

Name	Description	Type	Value
`QueueName`	The name to use for the Queue created. The Queue name must be unique for all queues created by the given Access Key ID.	Optional	An alphanumeric string specifying the name of the Queue to create. Maximum 80 characters.
`DefaultVisibilityTimeout`	Sets the default visibility timeout for this Queue. For more information, please see SetVisibilityTimeout.	Optional	Integer

Example CreateQueue Request (SOAP)

The following example of an `CreateQueue` action creates a new Queue named queue2.

Sample Request

```
<soap:Body wsu:Id="body"
 xm-
lns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-u
tility-1.0.xsd">
<m:CreateQueue xmlns:m="http://queue.amazonaws.com/doc/2006-04-01/">
<m:QueueName>queue5</m:QueueName>
</m:CreateQueue>
</soap:Body>
```

Sample Response

```
<CreateQueueResponse>
    <QueueUrl>
        http://queue.amazonaws.com/A29E9VSPHGOG23/queue2
    </QueueUrl>
    <ResponseStatus>
        <StatusCode>Success</StatusCode>
        <RequestId>
            5f461537-fe5d-45f0-b0e3-bd6d6ddea5d6
        </RequestId>
    </ResponseStatus>
</CreateQueueResponse>
```

Error Responses

Error	Description
InvalidParameterValue	QueueName did not pass validation.

Example Error Response

QueueName is too long or not alphanumeric results in the following error:

```
<soap:Envelope xmlns:soap=http://www.w3.org/2003/05/soap-envelope
 xmlns:aws=http://webservices.amazon.com/AWSFault/2005-15-09>
    <soap:Body>
        <soap:Fault>
            <soap:Code>
                <soap:Value>soap:Sender</soap:Value>
                <soap:Subcode>
                    <soap:Value>
                        aws:InvalidParameterValue
                    </soap:Value>
                </soap:Subcode>
            </soap:Code>
            <soap:Reason>
                <soap:Text xml:lang=en-US>
                    Value (abcdefghijklmnopqrstuvwxyzqueue5) for parameter
QueueName is invalid.
                    Must be an alphanumeric String of 1 to 20 in length
                </soap:Text>
            </soap:Reason>
            <soap:Detail>
                <aws:RequestId xm-
lns:aws=http://webservices.amazon.com/AWSFault/2005-15-09>
                    af8adcb5-2236-41e9-bdaf-8da6f218318c
                </aws:RequestId>
            </soap:Detail>
        </soap:Fault>
    </soap:Body>
</soap:Envelope>
```

ListQueues (SOAP)

The `ListQueues` action returns a list of the queues associated with the AWS account represented by the Access Key ID or X.509 certificate included in the request. A maximum of 10,000 queue URLs are returned. If a value is specified for the optional *`QueueNamePrefix`* parameter, only those queues with a queue name beginning with the value specified are returned. The queue name is specified in the *`QueueName`* parameter when a queue is created.

Validation

`ListQueues` requests do not contain any parameter values that require validation.

Returns

The `ListQueues` action, if usccessful, returns Success and a list of queue URLs owned by the AWS account represented by the access idetifiers included in the request. The queue URLs returned is determined by the Access Control policy set on the queue. It returns an error response is unsuccessful.

Request Parameters

The following table lists the parameters of the operation.

Name	Description	Type	Value
QueueNamePrefix	Optional. This parameter can be used to filter results returned. When specified, only queues with queue names beginning with the specified string are returned.	String	User-defined alphanumeric String

Example ListQueues Request (SOAP)

The following example of an `ListQueues` operation returns a list of queues owned by the sender of the request, as determined by the Access Key ID or certificate included in the request.

Sample Request

```
<soap:Body wsu:Id="body"
 xm-
lns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-u
tility-1.0.xsd">
  <ListQueuesRequest xmlns:sqs="http://queue.amazonaws.com/doc/2006-04-01/">
  </ListQueuesRequest>
</soap:Body>
```

Sample Response

The `ListQueues` action returns the Queue Resource (queueURL) associated with the Access Key ID or certificate used to authenticate the request.

```
<ListQueuesResponse>
    <QueueUrl>
        http://queue.amazonaws.com/A29E9VSPHGOG23/queueName
    </QueueUrl>
</ListQueuesResponse>
```

DeleteQueue (SOAP)

Deletes the queue specified by the Queue URL provided. A Queue is deleted only if it is does not contain any messages. Use DeleteMessage to delete messages from the queue.

Validation

`DeleteQueue` requests are validated on the following:

- The queue specified must exist.

Returns

This action returns either Success is successful, or an error response if unsuccessful.

Request Parameters

The `DeleteQueue` action takes no input parameters other than those common to all requests to SQS.

Example DeleteQueue Request (SOAP)

The following example of a `DeleteQueue` request deletes the queue specified by the provided queue URL.

Sample Request

```
<soap:Body wsu:Id="body"
 xm-
lns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-u
tility-1.0.xsd">
  <m:DeleteQueue xmlns:m="http://queue.amazonaws.com/doc/2006-04-01/">
  </m:DeleteQueue>
</soap:Body>
```

Sample Response

```
<DeleteQueueResponse>
    <ResponseStatus>
      <StatusCode>Success</StatusCode>
      <RequestId>cb919c0a-9bce-4afe-9b48-9bdf2412bb67</RequestId>
    </ResponseStatus>
</DeleteQueueResponse>
```

SendMessage (SOAP)

The `SendMessage` action delivers a message to the specified queue. A queue is represented by an queue URL. The content of the message is specified in the *`MessageBody`* parameter, and can be any alpha-numeric String.

For more information about URL encoding, please see http://en.wikipedia.org/wiki/URL_encoding.

Validation

`SendMessage` requests are validated on the following:

- *`MessageBody`* must exist and be a string of length 1 byte to 256K.

- The Queue specified must exist. The URL to the queue (the queue URL) must be a URL returned by a `CreateQueue` or `ListQueues` request.

Returns

This action, if successful, returns Success and a MessageId; a string representing the ID of the message sent. The message ID is used when performing actions on that message. This action returns an error response if unsuccessful.

Request Parameters

Name	Description	Type	Value
MessageBody	Required. A string representing the body of the message to send. The total string length of the message body can not exceed 256K.	An alpha-numeric String	The content of the message. The message can contain any valid string character, but must be URL-encoded so that the string does not contain any characters that would not be valid in an HTTP URL.

Example SendMessage Request (SOAP)

The following example `SendMessage` request sends a message containing "Your message text" as the body of the message to the specified queue.

Sample Request

```
<soap:Body wsu:Id="body"
 xm-
lns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-u
tility-1.0.xsd">
  <m:SendMessage xmlns:m="http://queue.amazonaws.com/doc/2006-01-01/">
    <m:MessageBodies>
      <m:MessageBody>
        Your message text
      </m:MessageBody>
    </m:MessageBodies>
  </m:SendMessage>
</soap:Body>
```

Sample Response

```
<SendMessageResponse>
    <MessageId>
        1EDR8H6XFYE9PGV3FGQQ|04WQYHSQ39E6Y1K6EDFS|0QE42ST4KW7RK9HSY074
    </MessageId>
</SendMessageResponse>
```

Error Response

Error	Description
MissingParameter	No value for *MessageBody* was supplied.
InvalidURI	The URL for the queue is not valid, or was not an URL returned by a CreateQueue or ListQueues request.

ReceiveMessage (SOAP)

Retrieves one or more messages from the queue specified, returning the message body and message ID of each message. Messages returned by this action stay in the queue until deleted. However, once a message is returned to a ReceiveMessage request, it will not be returned on subsequent ReceiveMessage requests until the duration of the *VisibilityTimeout* has passed. Please see SetVisibilityTimeout for more information.

> **Note**
>
> To retrieve a message from a queue regardless of visibility status, use PeekMessage.

Validation

ReceiveMessage action requests are validated on the following:

- The Queue specified must exist. The URL to the queue must be a queue URL returned from a CreateQueue or ListQueues request.
- VisibilityTimeout, if used, must be an integer between 0 and 86400.
- *NumberOfMessages*, if used, must be an integer between 1 and 255.

Returns

This action returns Success, the message ID and message body of each available message (those not restricted by the visibility timeout setting) if successful. It returns an error response if unsuccessful.

Request Parameters

Name	Description	Type	Value
NumberOfMessages	Optional. Specifies the maximum number of messages to return. If the number of messages in the queue is less than value specified by *NumberOfMessages*, then the number of messages returned is up to the number of messages in the queue. Not necessarily all the messages in the queue will	Integer	A value between 1 and 256.

Name	Description	Type	Value
	be returned. If no value is provided, the default value of 1 is used.		
VisibilityTimeout	Optional. The duration, in seconds, that the messages are visible in the queue. If no value is specified, the default value for the queue is used. The default value is set using `CreateQueue` or `SetVisibiltityTimeout`.	Integer	0 to 86400 seconds (maximum 24 hours)

Example ReceiveMessage Request (SOAP)

Sample Request

```
<soap:Body wsu:Id="body"
 xm-
lns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-u
tility-1.0.xsd">
  <m:ReceiveMessage xmlns:m="http://queue.amazonaws.com/doc/2006-01-01/">
  </m:ReceiveMessage>
</soap:Body>
```

Sample Response

```
<ReceiveMessageResponse>
    <Messages>
        <MessageId>
           17VXQHSGX0SG4ZEPPK7R|0QE42ST4KW7RK9HSY074|0Z4AN912X0H2EP8BV6XJ
        </MessageId>
        <MessageBody>foo</MessageBody>
    </Messages>
    <ResponseStatus>
      <StatusCode>Success</StatusCode>
      <RequestId>cb919c0a-9bce-4afe-9b48-9bdf2412bb67</RequestId>
    </ResponseStatus>
</ReceiveMessageResponse>
```

Error Responses

Error	Description
InvalidParameterValue	The value specified for *VisibilityTimeout* or *NumberOfMessages* is out of range.
InvalidURI	The URL for the queue is not valid, or is not an URL returned by a `CreateQueue` or `ListQueues` request.

DeleteMessage (SOAP)

The `DeleteMessage` action removes the specified message from the queue. Messages stay in the queue untill they are deleted with a `DeleteMessage` request.

Validation

`DeleteMessage` requests are validated on the following:

- The message specified must exist.

Returns

Success and the message ID of the deleted message if successful, or an error code if unsuccessful.

Request Parameters

Name	Description	Type	Value
MessageId	Required. The ID of the message to delete.	String	A message ID value returned from a SendMessage request.

Example DeleteMessage Request (SOAP)

The following example `DeleteMessage` request deletes a the message represented by the specified message.

Sample Request

```
<soap:Body wsu:Id="body"
 xm-
lns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-u
tility-1.0.xsd">
  <m:DeleteMessage xmlns:m="http://queue.amazonaws.com/doc/2006-01-01/">
    <m:MessageIds>
      <m:MessageId>
       0SM9BGMW4EBDZMX7FXD1|04WQYHSQ39E6Y1K6EDFS|0QE42ST4KW7RK9HSY074
      </m:MessageId>
    </m:MessageIds>
  </m:DeleteMessage>
</soap:Body>
```

Sample Response

```
<DeleteMessageResponse>
    <Message>
        <MessageId>
        0SM9BGMW4EBDZMX7FXD1|04WQYHSQ39E6Y1K6EDFS|0QE42ST4KW7RK9HSY074
        </MessageId>
    </Message>
    <ResponseStatus>
      <StatusCode>Success</StatusCode>
      <RequestId>cb919c0a-9bce-4afe-9b48-9bdf2412bb67</RequestId>
    </ResponseStatus>
</DeleteMessageResponse>
```

Error Response

Error	Description
MissingParameter	*MessageId* parameter was not present in the request.
InvalidURI	The URL for the queue is not valid, or was not an URL returned by a `CreateQueue` or `ListQueues` request.
InvalidParameterValue	No message exists with the ID specified.

PeekMessage (SOAP)

The `PeekMessage` action retrieves the messages specified by the message ID specified in the *MessageIds* parameter. The message is returned regardless of the *VisibilityTimeout* state on the queue. The visibility state is not modified when `PeekMessage` is used, thereby not affecting which messages get returned from a subsequent `ReceiveMessage` request.

Validation

A `PeekMessage` request is validated on the following:

- The Queue specified must exist. The URL to the queue must have be a queue URL returned from a `CreateQueue` or `ListQueues` request.
- MessageId must exist and be a value returned by a `ReceiveMessage` or `SendMessage` request.

Returns

This action returns Success, the message body, and the message ID of the message specified if successful. It returns an error response if unsuccessful.

Request Parameters

Name	Description	Type	Value
MessageId	Required. The message ID of the message to return.	String	17VXQHSGX0SG4ZEPPK7R|0QE42ST4KW7RK9HSY074|0Z4AN912X0H2EP8BV6XJ

Example PeekMessage Request (SOAP)

The following example `PeekMessage` request returns the message associated with the message ID specified.

Sample Request

```
<soap:Body wsu:Id="body"
 xm-
lns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-u
tility-1.0.xsd">
  <m:PeekMessage xmlns:m="http://queue.amazonaws.com/doc/2006-04-01/">
    <m:MessageId>
      <m:MessageId>
       17VXQHSGX0SG4ZEPPK7R%7C0QE42ST4KW7RK9HSY074%7C0Z4AN912X0H2EP8BV6XJ
      </m:MessageId>
    </m:MessageId>
  </m:PeekMessage>
</soap:Body>
```

Sample Response

```
<PeekMessageResponse>
    <Message>
        <MessageId>
        17VXQHSGX0SG4ZEPPK7R|0QE42ST4KW7RK9HSY074|0Z4AN912X0H2EP8BV6XJ
        </MessageId>
        <MessageBody>foo</MessageBody>
    </Message>
    <ResponseStatus>
      <StatusCode>Success</StatusCode>
      <RequestId>cb919c0a-9bce-4afe-9b48-9bdf2412bb67</RequestId>
    </ResponseStatus>
</PeekMessageResponse>
```

Error Response

Error	Description
MissingParameter	*MessageId* parameter was not present in the request.
InvalidURI	The URL for the queue is not valid, or was not an URL returned by a CreateQueue or ListQueues request.
InvalidParameterValue	No message exists with the message ID specified.

SetVisibilityTimeout (SOAP)

Sets the amount of time, *VisibilityTimeout*, messages are hidden from subsequent read requests after being retrieved by a ReceiveMessage request. The value is set in seconds, and measured from the time of the request.

To set the *VisibilityTimeout* on a select messages in the queue, specify the ID for that message using the *MessageId* parameter. If no message ID is provided, the *VisibilityTimeout* value is set for all messages in the queue.

The visibility timeout for a message indicates whether to return the message to a request. If a message is returned, it will not be returned on subsequent ReceiveMessage reqeust for the duration of the *VisibilityTimeout* period. Once that duration has passed, the message will again be returned to a ReceiveMessage request unless it is deleted from the queue.

Validation

The following criteria are validated in a `SetVisibilityTimeout` request.

- *`VisibilityTimeout`* must be an integer between 0 and 86400.
- The specified resource must exist (the URL to the resource must have been returned by an SQS request).

Returns

This action returns Success if successful, or an error code if unsuccessful.

Request Parameters

Name	Description	Type	Value
`VisibilityTimeout`	Required. The duration, in seconds, that the message specified by *`MessageIds`* is visible in the queue.	Integer	0 to 86400 seconds (maximum 24 hours)
`MessageIds`	Optional. Specifies a message ID for which to set the *`VisibilityTimeout`* value.	String	

Example SetVisibilityTimeout Request (SOAP)

The following example sets the visibility timeout to 35 seconds for all messages in the specified queue.

Sample Request

```
<soap:Body wsu:Id="body"
 xm-
lns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-u
tility-1.0.xsd">
    <m:SetVisibilityTimeout xm-
lns:m="http://queue.amazonaws.com/doc/2006-01-01/">
        <m:VisibilityTimeout>
            35
        </m:VisibilityTimeout>
    </m:SetVisibilityTimeout>
</soap:Body>
```

Sample Response

```
<SetVisibilityTimeoutResponse>
     <ResponseStatus>
      <StatusCode>Success</StatusCode>
      <RequestId>cb919c0a-9bce-4afe-9b48-9bdf2412bb67</RequestId>
```

```
    </ResponseStatus>
</SetVisibilityTimeoutResponse>
```

Error Responses

Error	Description
InvalidParameterValue	*VisibilityTimeout* was not an integer between 0 and 86400.
InvalidURI	The URL for the queue is not valid, or was not an URL returned by a CreateQueue or ListQueues request.

GetVisibilityTimeout (SOAP)

The GetVisibilityTimeout action retrieves the *VisibilityTimeout* value set on the queue specified.

The *VisibilityTimeout* is the amount of time for which a message is "invisible" to ReceiveMessage requests. For example, message A is in a queue and has a visibility timeout set at 30 seconds. A ReceiveMessage request is made on that queue, and Message A is returned. A subsequent ReceiveMessage reqeust is made in 10 seconds (within duration of *VisibilityTimeout* of 30 seconds). Message A is not returned to that request. Another request is made 20 seconds after the first request. Message A is again not returned. Any request made after 30 seconds has passed will return Message A, which resets the Visibility Timeout, until it is deleted from the queue.

Messages that are meant to be received only once should be deleted, using DeleteMessage within the duration of the *VisibilityTimeout*.

Returns

Returns Success if successful, or an error response if unsuccessful.

Request Parameters

This operation has no input parameters other than those common to all requests to SQS. Please see SOAP Parameters for more information.

Example GetVisibilityTimeout Request (SOAP)

The following example GetVisibilityTimeout request returns the *VisibilityTimeout* value for the queue specified.

Sample Request

```
<soap:Body wsu:Id="body"
 xm-
lns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-u
tility-1.0.xsd">
<m:GetVisibilityTimeout
xmlns:m="http://queue.amazonaws.com/doc/2006-04-01/">
```

```
</m:GetVisibilityTimeout>
</soap:Body>
```

Sample Response

```
<GetVisibilityTimeoutResponse>
    <VisibilityTimeout>
      35
    </VisibilityTimeout>
    <ResponseStatus>
      <StatusCode>Success</StatusCode>
      <RequestId>cb919c0a-9bce-4afe-9b48-9bdf2412bb67</RequestId>
    </ResponseStatus>
</GetVisibilityTimeoutResponse>
```

Error Responses

Error	Description
InvalidURI	The URL for the queue is not valid, or was not an URL returned by a `CreateQueue` or `ListQueues` request.

AddGrant (SOAP)

The `AddGrant` operation gives the grantee one or more permissions to perform actions on the the queue specified. The `AddGrant` action adds a grant to the specified resource, which adds permissions for the grantee to that queue.

Returns

The AddGrant request returns Success if the action succeeded, or an error response if unsuccessful.

Request Parameters

Name	Description	Type	Value
Action	Specifies the action to perform.	Required. String.	`AddGrant`
Version	The API version to use, as specified in the WSDL.	Required. Date.	2006-04-01
AWSAccessKeyId	The Access Key ID of the sender of the request.	Required. The AWS generated Access Key ID associated with the owner of the queue.	
Expires	The date and time at which the signature included in the request expires, in the format YYYY-	String	2006-07-07T15:04:56Z

Name	Description	Type	Value
	MM-DDThh:mm:ssZ, as specified in the ISO 8601 standard.		
Signature	A request signature is calculated by concatenating the values of each request parameter included in the request, in alphabetical order, and then calculating an RFC 2104-compliant HMAC-SHA1 hash, using the Secret AWS Access Key as the key. Please see the Request Authentication topic for more information about calculating signatures.	HMAC-SHA1 digest calculated from the request parameters.	Qn-pl4Qk/7tINHzfXCiT7VbBatDA=
QueueName	Specifies the queue for which the grant is applied.	Required. The name of the queue, as specified when the queue was created.	MyQueue
Grantee.EmailAddress	The email address of the user for which the grant is added.	Required. String.	email_alias@server_domain.com
Permission	The grant, or permission, to add for the Grantee. Please see ACS Permissions for a list of available grants.	String.	Read

Sample AddGrant Request (SOAP)

The following sample demonstrates adding a Read grant.

Sample Request

```
<soapenv:Body>
  <ac:AddGrant xmlns:ac="http://access.amazonaws.com/doc/2006-01-01/" xm-
lns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <ac:Request>
      <ac:Grantee xsi:type="ac:AmazonCustomerByEmail">
        <ac:EmailAddress>chriscustomer@email.com</ac:EmailAddress>
      </ac:Grantee>
      <ac:Permission>Read</ac:Permission>
    </ac:Request>
  </ac:AddGrant>
</soapenv:Body>
```

Sample Response

```
<soapenv:Body>
  <ac:AddGrantResponse xmlns:ac="http://queue.amazonaws.com/doc/2006-04-01/">
    <ac:Response>
      <ac:ResponseStatus>
        <ac:StatusCode>SUCCESS</ac:StatusCode>
        <ac:StatusMessage/>
      </ac:ResponseStatus>
    </ac:Response>
  </ac:AddGrantResponse>
</soapenv:Body>
```

ListGrants (SOAP)

The ListGrants action lists the grants (permissions) for the use of this queue. Only the owner of the queue can use the ListGrants action.

Any user calling any Access Control actions must have FullControl access to the specified queue for Access Control actions tol succeed. If the sender of the request does not have FullControl the operations will fail with an "Access Denied" error.

Returns

This action returns Success and a list of grants for the user specified if successful, or an error message if unsuccessful.

Request Parameters

Name	Description	Type	Value
Action	The action to perform.	String	ListGrants
Version	The API version to use, as specified in the WSDL.	Required. Date.	2006-04-01
AWSAccessKeyId	The Acccess Key ID of the owner of the queue.	Required. String	The Access Key ID of the sender of the request.
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.	String	2006-07-07T15:04:56Z
Signature	A request signature is calculated by concatenating the values of each request parameter included in the request, in alphabetical order, and then calculating an	HMAC-SHA1 digest calculated from the request parameters.	Qn-pl4Qk/7tINHzfXCiT7VbBatDA=

Name	Description	Type	Value
	RFC 2104-compliant HMAC-SHA1 hash, using the Secret AWS Access Key as the key. Please see the Request Authentication topic for more information about calculating signatures.		
QueueName	Specifies the queue for which the list of grants is returned.	Required. The name of the queue, as specified when the queue was created.	MyQueue
Grantee.EmailAddress	The email address of the entity to which acccess rights are to be granted.	String	email_alias@server_domain.com
Permission	If specified, the type of grant, or permission, to list for the specified Grantee. Please see ACS Permissions for a list of available permissions.	Optional. String.	Read

Sample ListGrants Request (SOAP)

This ListGrants SOAP request lists all the grants with Read permission for the user with the specfied email address

```
<soapenv:Body>
  <ac:ListGrants xmlns:ac="http://access.amazonaws.com/doc/2006-01-01/">
    <ac:Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <ac:Grantee xsi:type="ac:AmazonCustomerByEmail">
        <ac:EmailAddress>chriscustomer@email.com</ac:EmailAddress>
      </ac:Grantee>
      <ac:Permission>Read</ac:Permission>
    </ac:Request>
  </ac:ListGrants>
</soapenv:Body>
```

If the Permission element is omitted, all the grants for a user are listed. This `ListGrants` SOAP request lists all the grants for the user identified by the specified Canonical User ID.

```
<soapenv:Body>
  <ac:ListGrants xmlns:ac="http://access.amazonaws.com/doc/2006-01-01/">
    <ac:Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

```
        <ac:Grantee xsi:type="ac:CanonicalUser">
<ac:ID>93ddb081b6ca9f35fbf3626dde06b73854d43b9a7b6305201550c90aa7bd1eef</ac:I
D>
        </ac:Grantee>
      </ac:Request>
    </ac:ListGrants>
  </soapenv:Body>
```

If both the Permission and the Grantee are omitted, all the grants for all the users are listed. This `ListGrants` query request lists all the grants on the queue specified.

```
  <soapenv:Body>
    <ac:ListGrants xmlns:ac="http://access.amazonaws.com/doc/2006-01-01/">
      <ac:Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      </ac:Request>
    </ac:ListGrants>
  </soapenv:Body>
```

Sample Response

```
<ListGrantsResponse xmlns="http://access.amazonaws.com/doc/2006-01-01/">
  <GrantList>
    <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="CanonicalUser">
<ID>79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be>/ID>
      <DisplayName>filesinc>/DisplayName>
    </Grantee>
    <Permission>ReceiveMessage>/Permission>
  </GrantList>
  <GrantList>
    <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="CanonicalUser">
<ID>a9a7b886d6fd24a52fe8ca5bef65f89a64e0193f23000e241bf9b1c61be666e9>/ID>
      <DisplayName>chriscustomer>/DisplayName>
    </Grantee>
    <Permission>FullControl>/Permission>
  </GrantList>
  <ResponseStatus>
    <StatusCode>Success>/StatusCode>
  </ResponseStatus>
</ListGrantsResponse>
```

RemoveGrant (SOAP)

The `RemoveGrant` action revokes a grant (permission) for the use of the specified queue. Only the owner of the queue, or grantee with Full Control permissions for the queue, can add or remove grants.

Returns

Returns Success if successful, or an error response if unsuccessful.

Request Parameters

The following table lists the parameters for the RemoveGrant operation.

Name	Description	Type	Value
Action	Specifies the action (or operation) to perform.	Required. String.	RemoveGrant
Version	The API version to use, as specified in the WSDL.	Required. Date.	2006-04-01
AWSAccessKeyId	Your Access Key ID	Required. The AWS generated Access Key ID associated with the owner of the queue.	
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.	String	2006-07-07T15:04:56Z
Signature	A request signature is calculated by concatenating the values of each request parameter included in the request, in alphabetical order, and then calculating an RFC 2104-compliant HMAC-SHA1 hash, using the Secret AWS Access Key as the key. Please see the Request Authentication topic for more information about calculating signatures.	HMAC-SHA1 digest calculated from the request parameters.	Qn-pl4Qk/7tINHzfXCiT7VbBatDA=
QueueName	Specifies the queue to which the grant is applied.	Required. The name of the queue, as specified when the queue was created.	
Grantee.EmailAddress	The email address of the user for which the grant is added.	Required. String.	email_alias@server_domain.com
Permission	The grant, or permission, to remove from the Grantee. Please see ACS Permissions for a list of available grants.	String.	Read

Example RemoveGrant Request (SOAP)

The following example `RemoveGrant` request removes all access rights for the queue named *queue2* from the user with the specified email address.

Sample Request

```
<soapenv:Body>
  <ac:RemoveGrant xmlns:ac="http://access.amazonaws.com/doc/2006-01-01/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <ac:Request>
      <ac:Grantee xsi:type="ac:AmazonCustomerByEmail">
        <ac:EmailAddress>chriscustomer@email.com</ac:EmailAddress>
      </ac:Grantee>
      <ac:Permission>Read</ac:Permission>
    </ac:Request>
  </ac:RemoveGrant>
</soapenv:Body>
```

SOAP Faults

The following table lists the SOAP faults that are returned by the Simple Queue Service.

Fault	Reason	Sample Response

InvalidSecurity, Request Has Expired	The time to live set on the request has expired	`<soap:Fault>`
		`<soap:Code>`
		`<soap:Value>soap:Sender</soap:Value>`
		`<soap:Subcode>`
		`<soap:Value>aws:InvalidSecurity</soap:Value>`
		`</soap:Subcode>`
		`</soap:Code>`
		`<soap:Reason>`
		`<soap:Text xml:lang=en-US>`
		`Request has expired`
		`</soap:Text>`
		`</soap:Reason>`
		`<soap:Detail>`
		`<aws:RequestId xmlns:aws=http://webservices.amazon.com/AWSFault/2005-15-09>`
		`0e46d0ca-a0ab-4730-9bca-219f064e16fd`
		`</aws:RequestId>`
		`</soap:Detail>`
		`<soap:Fault>`

SQS REST Reference

The Amazon Simple Queue Service supports the following actions in REST requests:

- Common REST Parameters
- REST Headers
- POST on QueueService
- GET on QueueService
- PUT on MessageQueue
- GET on MessageQueue
- DELETE on MessageQueue
- GET on Message
- DELETE on Message
- REST Errors

All actions return a ResponseStatus element that contains the following elements:

- RequestId: an ID that uniquely identifies this request
- Status: String, one of Success, Warnings, or Errors
- Warnings: Array of (String code, String message) if there were any warnings
- Errors: Array of (String code, String message) if there were any errors

Common REST Parameters

The following parameters must be included in each REST request to SQS:

Parameter Name	Description	Example Value
Action	Indicates the action to perform.	GET
Header	Please see REST Headers for more information.	ContentType

When passing these values in on the address line of a browser they must be url-encoded. This is true for any parameter passed to SQS using a browser and is typically necessary in the Signature, MessageIds, and MessageBody fields.

REST HTTP Headers

The following headers are used in REST requests to SQS.

`Authorization Header`

A string of the following format: `<AWSAccessKeyId>:<Signature>`. The Signature is an HMAC-SHA1 hash of the string:

```
HTTP-METHOD>\n<content-MD5>\n<ContentType>\n<date>\n<path>
```

For example:

```
PUT\n\ntext/plain\nThu, 01 Jun 2006 12:12:23 PDT\n/
```

Once the signature is computed using the AWS Secret Key and the above string the Authorization header should have the following format:

```
AWS <awsAccessKeyId>:<signature>
```

where the space is required after 'AWS'.

`Other required headers:`

- Version: *AWS-Version*, in this case 2006-04-01
- Date: *date*, the same date used to create the signature as previously described.
- Content Type: *content-type*, the same content type used to create the string to sign.

`Optional header:`

- *content-md5*, if used to compute the string to sign, the value of the content-md5 header is not compared with the actual md5 computed from the received message.

Please see the Authenticating REST Requests topic for more information.

Once the signature is computed using the AWS Secret Access Key and the preceding string, the Authorization header should have the following format (note the required space after 'AWS'):

```
AWS <awsAccessKeyId>:<signature>
```

POST on QueueService

The `POST` action creates a new queue. An optional queue name may be provided to associate with the queue for future reference, but the queue name must be unique among the queues associated with the AWS account associated with the Access Key ID provided in the request. All queues are assigned a queue ID, which is unique to each AWS account signed up for SQS. The URL to the queue, the Queue URL, is used to specify the queue on which to perform actions.

If no value for *`QueueName`* is provided, a unique string is generated for the queue name.

If the queue already exists, `POST` on QueueService returns the Queue URL with a error indicating that the queue already exists.

When a queue is created, Full Control access rights are granted to creator of the queue. Only the owner of the queue , or grantee assigned Full Control rights to the queue, can grant or deny access rights to the queue and messages in it.

> **Note**
>
> The AWS account associated with the Access Key ID must be signed up for Simple Queue Service for the request to succeed, as Simple Queue Service is not a free service.

The default value for `VisibilityTimeout` is also set when a queue is created. Visibility assists in assuring that messages are only delivered once. `VisibilityTimeout` refers to the length of time, in seconds, that a message will not be returned to a request to retrieve it after already being returned to a similar request. In other words, if a message is retrieved from a queue, it will not be retrieved again from another request for messages in that queue until the druation of `VisibilityTimeout` has passed. This value applies to all messages in the queue, unless otherwise specified by another action on the message.

Returns

Returns Success and a queue URL if succeessful, or an error response if unsuccessful.

Example POST Request on QueueService

The following example `POST` request on the QueueService endpoint creates a queue named "MyQueue".

Sample Request

```
POST /?QueueName=MyQueue HTTP/1.1
Host: queue.amazonaws.com
Authorization: [AWS authentication string]
Content-MD5: ...
Content-Length: 0
Date: Wed, 05 Apr 2006 21:12:00 GMT
```

Sample Response

```
HTTP/1.1 200
Date: Wed, 05 Apr 2006 21:12:00 GMT
Content-Length: 234

<?xml version="1.0" encoding="UTF-8"?>
<CreateQueueResponse xmlns:sqs=http://queue.amazonaws.com/doc/2006-04-01/>
        <QueueUrl>
            http://queue.amazonaws.com/A23E9WXPHGOG29/MyQueue
        </QueueUrl>
</CreateQueueResponse>
```

GET Action on QueueService

The `GET` action returns a list of the queue URLs. This action is used to list the existing queues associated with the AWS account represented by the Access Key ID included in the request. A maximum of 10,000 queue URLs are returned. If a value is specified for the optional `QueueNamePrefix` parameter, only those queues with a queue name beginning with the value specified are returned. The queue name is specified in the `QueueName` parameter when a queue is created.

Validation

`GET` requests on QueueService do not contain any parameter values that require validation.

Returns

The `GET` action on QueueService returns a list of Queue URLs representing the queues associated with the AWS account for the Access Key ID included in the request.

Request Parameters

The following table lists the parameters of the GET action on QueueService.

Name	Description	Type	Value
`QueueNamePrefix`	An optional parameter that you can use to filter results returned. When specified, only queues with queue names beginning with the specified string are returned.	String [Optional]	User-defined alphanumeric String

Example GET Request on QueueService

The following example `GET` request returns the queues with a name that begins with "Test".

`Sample Request`

```
GET /?QueueNamePrefix=Test HTTP/1.1
Host: queue.amazonaws.com
Authorization: [AWS authentication string]
AWS-Version:2006-04-01
Content-MD5: ...
Content-Length: 0
Date: Wed, 05 Apr 2006 21:12:00 GMT
```

`Sample Response`

```
HTTP/1.1 200
Date: Wed, 05 Apr 2006 21:12:00 GMT
Content-Length: 234

<?xml version="1.0" encoding="UTF-8"?>
<ListQueuesResponse xmlns:sqs=http://queue.amazonaws.com/doc/2006-04-01/>
        <QueueUrl>
            http://queue.amazonaws.com/QueueID/TestQueueAlpha
        </QueueUrl>
        <QueueUrl>
            http://queue.amazonaws.com/QueueID/TestQueueBeta
        </QueueUrl>
</ListQueuesResponse>
```

DELETE on MessageQueue

Deletes the queue specified by the QueueId. The QueueId for a queue is returned when a queue is created. A Queue is deleted only if it is does not contain any messages. Use DELETE on Message to delete messages from the queue.

Validation

`DELETE requests on a MessageQueue endpoint` are validated on the following:

- The queue specified must exist.

Returns

This action returns 200 (success) if successful, or 400 (fail) if unsuccessful.

Request Parameters

This action takes no input parameters other than those common to all REST requests.

Example DELETE Request on MessageQueue

The following example of a `DELETE` action deletes the queue represented by the QueueId provided.

Sample Request

```
DELETE /QueueId HTTP/1.1
Host: queue.amazonaws.com
Authorization: [AWS authentication string]
Content-MD5: ...
Content-Length: 0
Date: Wed, 05 Apr 2006 21:12:00 GMT
```

Sample Response

```
HTTP/1.1 200
Date: Wed, 05 Apr 2006 21:12:00 GMT
Content-Length: 0
```

GET on MessageQueue

Retrieves one or more messages from the queue specified, returning the message body and message ID of each message. Messages are retrieved form the front of the queue, and stay in the queue until deleted. However, once a message is returned to a `GET` request on a MessageQueue endpoint, it will not be returned on subsequent requests until the duration of the *`VisibilityTimeout`* has passed. Please see SetVisibilityTimeout for more information.

Validation

GET requests on MessageQueue are validated on the following:

- The Queue specified must exist. The URL to the queue must have been returned by a `POST` or `GET` on the QueueService endpoint.
- VisibilityTimeout, if used, must be an integer between 0 and 86400.
- *`NumberOfMessages`*, if used, must be an integer between 1 and 255.

Returns

This action returns the message body of each available (those not restricted by the visibility timeout setting) message if successful. It returns 400 (fail) and an error if unsuccessful.

Request Parameters

Name	Description	Type	Value
`MessageId`		String	
`VisibilityTimeout`	Optional. The duration, in seconds, that the messages are visible in the queue. If no value is specified, the default value for the queue is used. The default value is set using `CreateQueue` or `SetVisibiltityTimeout`.	Integer	0 to 86400 seconds (maximum 24 hours)
`NumberOfMessages`	Optional. Specifies the maximum number of messages to return. If the number of messages in the queue is less than value specified by *`NumberOfMessages`*, then the number of messages returned is up to the number of messages in the queue. Not necessarily all the messages in the queue will be returned. If no value is provided, the default value of 1 is used.	Integer	A value between 1 and 256.

Example GET Request on MessageQueue

The following example of a `GET` request on a MesageQueue endpoint returns up to 2 messages from the queue.

Sample Request

```
GET /QueueId/front?VisibilityTimeout=35&NumberofMessages=2 HTTP/1.1
Date: Wed, 05 Apr 2006 21:12:00 GMT
Content-Length: 24
Content-Type: text/plain
```

Sample Response

```
HTTP/1.1 200
Date: Wed, 05 Apr 2006 21:12:00 GMT
Content-Length: 24
Content-Type: text/plain
<Message>
Message Body A
</Message>
<Message>
Message Body B
</Message>
```

PUT on MessageQueue

The PUT action on a MessageQueue endpoint sends a message to the specified queue. A queue is represented by a Qqueue URL. The message is sent to the back of the queue, as messages are read from the front of the queue. The content of the message is specified in the *Message* parameter, and can be any String. The total string length of *Message* cannot exceed 256K.

Validation

PUT requests on a MessageQueue are validated on the following:

- *Message* must exist and be a string of length 1 byte to 256K.
- The Queue specified must exist. The URL to the queue must have been returned by a POST or GET request on hte QueueService endpoint.

Returns

This action, if successful, returns Success and a string representing the ID of the message sent, which is used as the value of the *MessageId* parameter when performing actions on that message. This action returns an error response if unsuccessful.

Request Parameters

Name	Description	Type	Value
Message	Specifies the body of the message to send. The total string length of all the message bodies can not exceed 256K.	Required	String

Example PUT Request on MessageQueue

The following example PUT request on a MessageQueue endpoint sends a message to the queue specified by the

Sample Request

```
PUT /QueueId/back HTTP/1.1
Date: Wed, 05 Apr 2006 21:12:00 GMT
Content-Length: 24
Content-Type: text/plain
<Message>Message Text Here</Message>
```

Sample Response

```
HTTP/1.1 200
Date: Wed, 05 Apr 2006 21:12:00 GMT
Content-Length: 24
Content-Type: text/plain
<MessageId>
17VXQHSGX0SG4ZEPPK7R|0QE42ST4KW7RK9HSY074|0Z4AN912X0H2EP8BV6XJ
</MessageId>
```

DELETE on Message

The DELETE action on a Message endpoint removes the specified message from the queue. Messages stay in the queue until they are deleted with a DELETE request.

Validation

DELETE requests are validated on the following:

- The message specified must exist.

Note

Even if the message is locked by another reader due to the Visibility Timeout setting it will still be deleted from the queue.

Returns

This action returns Succes if successful, or an error if unsuccessful.

Request Parameters

Name	Description	Type	Value
messageId	Required. The ID of the message to delete.	String.	The messageId returned from a PUT or GET request on a Messae-

Name	Description	Type	Value
			gQueue.

Example DELETE Request on a Message

The following example of a `DELETE` action removes the message specified from the queue.

Sample Request

```
DELETE /QueueId/MessageId HTTP/1.1
Host: queue.amazonaws.com
Authorization: [AWS authentication string]
Content-MD5: ...
Content-Length: 0
Date: 2007-01-12T12:00:00Z
```

Sample Response

```
HTTP/1.1 200
Date: Wed, 05 Apr 2006 21:12:00 GMT
Content-Length: 0
```

GET on Message

Retrieves the message specified from the queue, returning the body of the message. Messages returned by this action stay in the queue until deleted, and the VisibilityTimeout is not affected. Please see SetVisibilityTimeout for more information.

Validation

GET requests on a message are validated on the following:

- The message specified must exist. The URL to the message must have been returned by a `PUT` or `GET` request on a MessageQueue.

Returns

This action returns the body of the message specified, if it exists.

Request Parameters

Name	Description	Type	Value
MessageId	The ID of the message to delete.	Required. String.	17VXQHSGX0SG4ZEPPK7R%7C0QE42ST4KW7RK9HSY074%7C0Z4AN912X0H2EP8BV6XJ

Example GET Request on a Message

The following example of a *GET* action on a Message returns the message specified by the message ID.

```
GET /QueueId/MessageId HTTP/1.1
Host: queue.amazonaws.com
Authorization: [AWS authentication string]
Content-MD5: ...
Content-Length: 0
Date: Wed, 05 Apr 2006 21:12:00 GMT
```

Sample Response

```
HTTP/1.1 200
Date: Wed, 05 Apr 2006 21:12:00 GMT
Content-Length: 24
Content-Type: text/plain
<Message>
Message body
</Message>
```

Common REST Errors

The following list includes the error codes that can be returned from REST requests:

Error	Reason	Sample Response

MissingParameter	A required parameter is missing	`<Response>`
		`<Errors>`
		`<Error>`
		`<Code>MissingParameter</Code>`
		`<Message>`
		`The request has a missing parameter Action`
		`</Message>`
		`</Error>`
		`</Errors>`
		`<awsRequestId>`
		`427cce5e-950b-4f82-b30d-d232796a81b1`
		`</awsRequestId>`
		`<MissingParameterName>`
		`Action`
		`</MissingParameterName>`
		`</Response>`

AccessFailure	Some access credential could not be validated, such as Signature	`<Response>`
		`<Errors>`
		`<Error>`
		`<Code>AccessFailure</Code>`
		`<Message>`
		`AWS was not able to grant access to the QueueService service`
		`</Message>`
		`</Error>`
		`</Errors>`
		`<awsRequestId>`
		`24f372aa-f7bc-41e4-ad7b-5ec24be5b2f1`

Error	Reason	Sample Response

		`    </awsRequestId>`
		`    <ServiceName>QueueService</ServiceName>`
		`</Response>`

InvalidParameterCombination	Two parameters were specified that could not be used together, such as Timestamp and Expires	
ServiceUnavailable	Some required server needed by SQS is unavailable. This error is often temporary and the request be resent after a short wait.	
InternalError	There is an internal problem with SQS which cannot be resolved by the sender.	