
Amazon Simple Queue Service

入门指南

API Version 2012-11-05



Amazon Web Services

Amazon Simple Queue Service: 入门指南

Amazon Web Services

Copyright © 2013 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

The following are trademarks or registered trademarks of Amazon: Amazon, Amazon.com, Amazon.com Design, Amazon DevPay, Amazon EC2, Amazon Web Services Design, AWS, CloudFront, EC2, Elastic Compute Cloud, Kindle, and Mechanical Turk. In addition, Amazon.com graphics, logos, page headers, button icons, scripts, and service names are trademarks, or trade dress of Amazon in the U.S. and/or other countries. Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon.

All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

欢迎	1
Amazon SQS 简介	2
开始设置	6
创建 AWS 账户	6
注册 Amazon SQS	7
获取访问密钥 ID 和私有访问密钥	8
获取您需要的工具	9
使用 Amazon SQS	10
准备示例	10
创建队列	12
确认队列存在	14
发送消息	16
接收消息	19
删除消息	25
您已完成！	28
我从这里可以继续进行哪些内容？	29
文档历史记录	31

欢迎

Topics

- [Amazon SQS 资源 \(p. 1\)](#)

欢迎访问 *Amazon Simple Queue Service Getting Started Guide*。Amazon Simple Queue Service 是一项消息队列服务：该服务处理系统中其他组件间的消息或工作流。

以下视频向您演示了本指南中介绍的示例：[Getting Started with Amazon SQS](#)。

Amazon SQS 资源

您会发现下列相关资源在使用此服务的过程中会有所帮助。

[Amazon Simple Queue Service 开发者指南](#)

此开发人员指南详细探讨了该服务，包括一篇结构性概述和一份编程参考。

[Amazon Simple Queue Service API 参考](#)

API 参考提供了 WSDL 位置；API 操作、参数和数据类型的完整描述；该服务返回的错误列表

[使用 Amazon SQS 队列来确定何时进行自动扩展](#)

可以使用 Amazon SQS 队列来帮助确定应用程序的负载，并且在结合使用 Auto Scaling 时，可以根据流量大小对 Amazon EC2 实例的数量进行缩放。

[Amazon SQS 发行说明](#)

发行说明从高层面上概括介绍了当前发行版本的情况。特别说明了一些新功能、修复和已知问题。

[产品信息 Amazon SQS](#)

提供 Amazon SQS 相关信息的主要 Web 页面。

[Amazon SQS 开发论坛](#)

由开发人员组成的社区形式的论坛，他们可以在这里讨论与 Amazon SQS 有关的技术问题。

[AWS Premium Support Center](#)

提供 AWS Premium Support 相关信息的主要 Web 页面，这是一种一对一、快速响应的支持渠道，能帮助您您在 AWS 基础设施服务上构建和运行应用程序。

Amazon SQS 简介

Topics

- [Amazon SQS 概述 \(p. 2\)](#)
- [特色 \(p. 3\)](#)
- [消息生命周期 \(p. 3\)](#)
- [关于示例 \(p. 5\)](#)

本 Amazon SQS 简介意在为您提供该 Web 服务的宏观概述。阅读本部分后，您应理解练习本指南中各个示例所需的基本原理。

Amazon SQS 概述

Amazon SQS is a distributed queue system that enables web service applications to quickly and reliably queue messages that one component in the application generates to be consumed by another component. A queue is a temporary repository for messages that are awaiting processing.

Using Amazon SQS, you can decouple the components of an application so they run independently, with Amazon SQS easing message management between components. Any component of a distributed application can store messages in a fail-safe queue. Messages can contain up to 256 KB of text in any format. Any component can later retrieve the messages programmatically using the Amazon SQS API.

The queue acts as a buffer between the component producing and saving data, and the component receiving the data for processing. This means the queue resolves issues that arise if the producer is producing work faster than the consumer can process it, or if the producer or consumer are only intermittently connected to the network.

Amazon SQS ensures delivery of each message at least once, and supports multiple readers and 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.

Amazon SQS is engineered to always be available and deliver messages. One of the resulting tradeoffs is that SQS does not guarantee first in, first out delivery of messages. For many distributed applications, each message can stand on its own, and as long as all messages are delivered, the order is not important. If your system requires that order be preserved, you can place sequencing information in each message, so that you can reorder the messages when the queue returns them.

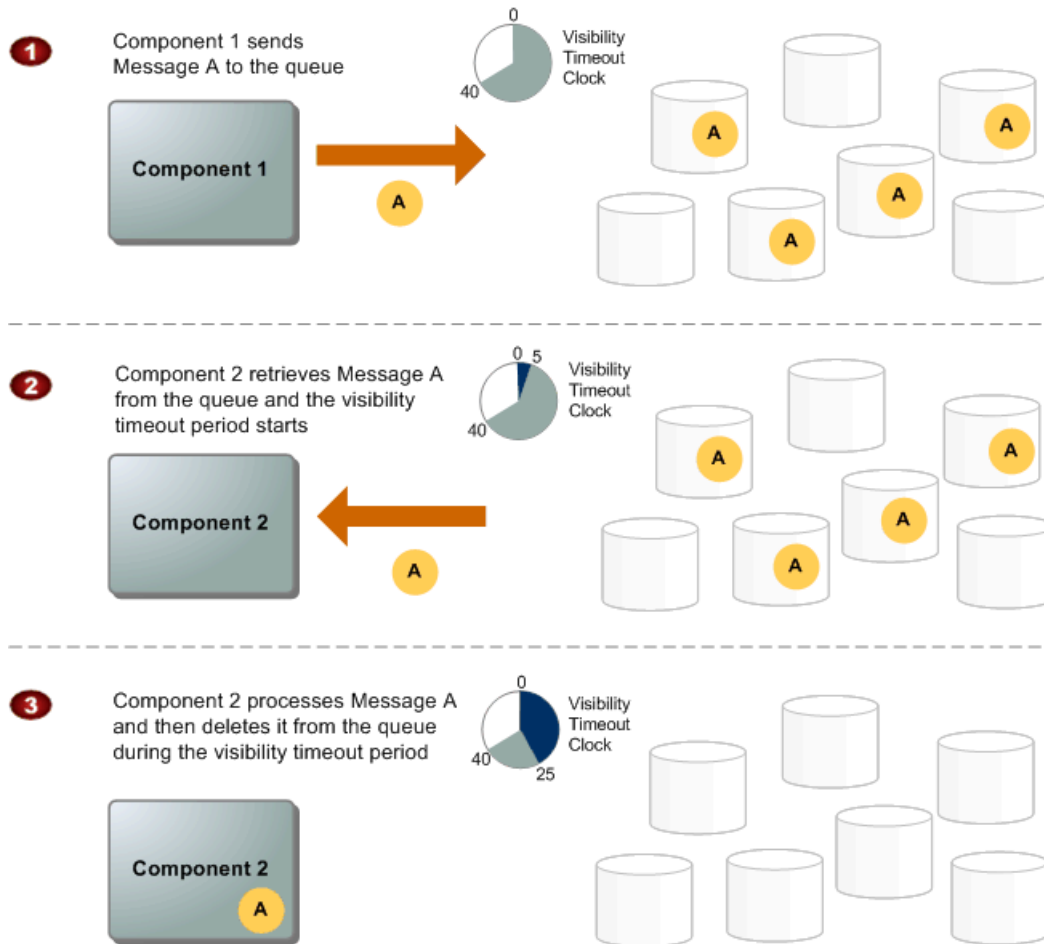
特色

Amazon SQS provides the following major features:

- Redundant infrastructure – Guarantees delivery of your messages at least once, highly concurrent access to messages, and high availability for sending and retrieving messages
- Multiple writers and readers – Multiple parts of your system can send or receive messages at the same time
Amazon SQS locks the message during processing, keeping other parts of your system from processing the message simultaneously.
- Configurable settings per queue – All of your queues don't have to be exactly alike
For example, one queue can be optimized for messages that require a longer processing time than others.
- Variable message size – Your messages can be up to 262,144 bytes (256 KB) in size
For even larger messages, you can store the contents of the message using the Amazon Simple Storage Service (Amazon S3) or Amazon SimpleDB, and use Amazon SQS to hold a pointer to the Amazon S3 or Amazon SimpleDB object. Alternately, you can split the larger message into smaller ones.
For more information about the services, go to the [Amazon S3 detail page](#) and the [Amazon SimpleDB detail page](#).
- Access control – You can control who can send messages to a queue, and who can receive messages from a queue
- Delay Queues – A delay queue is one which the user sets a default delay on a queue such that delivery of all messages enqueued will be postponed for that duration of time. You can set the delay value when you create a queue with `CreateQueue`, and you can update the value with `SetQueueAttributes`. If you update the value, the new value affects only messages enqueued after the update.

消息生命周期

The following diagram and process describe the lifecycle of an Amazon SQS message, called *Message A*, from creation to deletion. Assume that a queue already exists.



Message Lifecycle

1	Component 1 sends Message A to a queue, and the message is redundantly distributed across the SQS servers.
2	When Component 2 is ready to process a message, it retrieves messages from the queue, and Message A is returned. While Message A is being processed, it remains in the queue and is not returned to subsequent receive requests for the duration of the <i>visibility timeout</i> .
3	Component 2 deletes Message A from the queue to avoid the message being received and processed again once the visibility timeout expires.



Note

SQS automatically deletes messages that have been in a queue for more than maximum message retention period. The default message retention period is 4 days. However, you can set the message retention period to a value from 60 seconds to 1209600 seconds (14 days) with [SetQueueAttributes](#).

关于示例

在上一部分中，我们大体讨论了您的系统如何建立队列、确认队列已可供使用以及如何开始使用。使用期间，您的系统的各个组件不断发送、接收和删除消息。随本指南提供的示例库文件涵盖了可对 Amazon SQS 执行的全部操作。不过，本指南中的示例特别着重于核心队列操作：

- 创建队列
- 列出队列
- 向队列发送消息
- 从队列检索消息
- 从队列删除消息

有关示例的更多具体信息，请参阅[准备示例 \(p. 10\)](#)。

有关可对 Amazon SQS 执行的其他操作的信息，请参阅[Amazon Simple Queue Service 开发者指南](#)

开始设置

本部分引导您完成在提出 Amazon SQS 请求前必须完成的各个任务。这些任务按最佳顺序排列，便于您尽快运行示例。下表显示了您成为 AWS 用户或首次接触 AWS 时需要阅读的部分。

现有 AWS 用户	新 AWS 用户
<ol style="list-style-type: none">1. 注册 Amazon SQS (p. 7)2. 获取您需要的工具 (p. 9)	<ol style="list-style-type: none">1. 创建 AWS 账户 (p. 6)2. 注册 Amazon SQS (p. 7)3. 获取访问密钥 ID 和私有访问密钥 (p. 8)4. 获取您需要的工具 (p. 9)

创建 AWS 账户



Tip

如果您已是 AWS 的用户，可以直接跳到 [注册 Amazon SQS \(p. 7\)](#)。

您必须首先在 <http://aws.amazon.com> 上创建 AWS 账户，才能访问 AWS 提供的任何 Web 服务。AWS 账户只是可以使用 AWS 产品的 Amazon.com 账户；您可以在创建 AWS 账户时使用现有的 Amazon.com 账户登录名和密码。



Important

如果您有 Amazon.com 个人账户，可能希望一个仅用于 AWS 活动的独立 Amazon.com 账户。您可以提供 Amazon.com 系统尚未记录的新电子邮箱地址，也可以为您现有的 Amazon.com 账户提供一个电子邮箱地址，但使用不同的密码。您可以拥有使用电子邮箱地址相同但密码不同的多个 Amazon.com 账户。

您可以从您的 AWS 账户查看 AWS 账户活动和使用率报告，以及管理 AWS 账户访问标识符。

如需设置新账户

1. 转到 <http://aws.amazon.com>。

2. 在 请注册 AWS 框中，单击 立即注册。
登录 页面随即显示出来。
3. 输入有效的电子邮箱地址，选择 不，我是新客户 按钮，然后单击 继续。
下一页面要求您提供希望与该账户关联的名称（例如：Joe Smith）和账户密码。如果使用的是 Amazon.com 系统中已存在的电子邮箱地址，页面会显示该地址已存在，但仍然让您使用该地址创建新账户。
4. 输入您希望与账户关联的名称以及密码，然后单击 继续。
帐户信息 页面随即显示出来。
5. 输入您的联系信息并选择了解到 AWS 的方式。阅读 AWS 客户协议，选中相应的复选框以表明您已阅读并同意客户协议的条款，然后单击 继续。
该过程至此完成，您已成功创建新 AWS 账户。

此时，您已拥有 AWS 账户，但尚未注册使用 Amazon SQS。有关说明，请参阅[注册 Amazon SQS \(p. 7\)](#)。

注册 Amazon SQS

要使用 Amazon SQS，必须首先注册一个 AWS 账户（假设您尚无此账户）。如果您已有 AWS 账户，请继续并跳到[获取您需要的工具 \(p. 9\)](#)部分。如果您首次接触 AWS，请转到[获取访问密钥 ID 和私有访问密钥 \(p. 8\)](#)。否则，请按照以下步骤操作。

如需注册 AWS 账户

1. 请转至 <http://aws.amazon.com>，然后单击“注册”。
2. 按照屏幕上的说明进行操作。

在注册过程中，您会接到一个电话，需要您使用电话按键输入 PIN 码。

获取访问密钥 ID 和私有访问密钥

To get your access key ID and secret access key

Access keys consist of an access key ID and secret access key, which are used to sign programmatic requests that you make to AWS. If you don't have access keys, you can create them by using the AWS Management Console. We recommend that you use IAM access keys instead of AWS root account access keys. IAM lets you securely control access to AWS services and resources in your AWS account.



Note

To create access keys, you must have permissions to perform the required IAM actions. For more information, see [Granting IAM User Permission to Manage Password Policy and Credentials](#) in *使用 IAM*.

1. Go to the [IAM console](#).
2. From the navigation menu, click Users.
3. Select your IAM user name.
4. Click User Actions, and then click Manage Access Keys.
5. Click Create Access Key.

Your keys will look something like this:

- Access key ID example: AKIAIOSFODNN7EXAMPLE
- Secret access key example: wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY

6. Click Download Credentials, and store the keys in a secure location.

Your secret key will no longer be available through the AWS Management Console; you will have the only copy. Keep it confidential in order to protect your account, and never email it. Do not share it outside your organization, even if an inquiry appears to come from AWS or Amazon.com. No one who legitimately represents Amazon will ever ask you for your secret key.

Related topics

- [What Is IAM?](#) in *使用 IAM*
- [AWS Security Credentials](#) in *AWS General Reference*

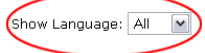
获取您需要的工具

如果想使用此指南中的示例代码，必须安装本部分中列出的编程工具。



Tip

在本文件的 HTML 版本中，您可以隐藏本指南中不适用于您所用编程语言的部分。在含有语言相关文本的页面的右上角，有一个语言选择菜单。选择您的语言以隐藏所有其他语言，或者选择“全部”，显示所有可用语言的示例。



您的选择会应用到本指南中的所有页面。

Java

您必须拥有以下工具才能使用 Java 示例代码：

- [Java 标准版开发工具包](#)

C#

您必须拥有以下工具才能使用 C# 示例代码：

- Microsoft Visual C# 2008 Express Edition 或 Microsoft Visual Studio 2008 Professional Edition
- .NET Framework 2.0 或更新版本

PHP 5

您必须拥有以下工具才能使用 PHP 示例代码：

- PHP 5.2.1 或更新版本（有关更多信息，请访问 <http://php.net>）
- AWS SDK for PHP（有关更多信息，请访问 <http://aws.amazon.com/sdkforphp/>）

您还需：

- 按照“Getting Started with the AWS SDK for PHP”<http://aws.amazon.com/articles/4261> 一文中“Configure the SDK Security Credentials”部分的说明，将证书添加到您的 config.inc.php 文件
- 验证您的 PHP 环境是否通过 SDK 兼容性测试 `sdk_compatibility_test.php`，该文件位于 AWS SDK for PHP 下载文件的 `_compatibility_test` 子目录中。

使用 Amazon SQS

Topics

- [准备示例 \(p. 10\)](#)
- [创建队列 \(p. 12\)](#)
- [确认队列存在 \(p. 14\)](#)
- [发送消息 \(p. 16\)](#)
- [接收消息 \(p. 19\)](#)
- [删除消息 \(p. 25\)](#)

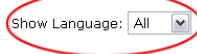
本部分介绍如何创建 Amazon SQS 队列、向队列发送消息，以及如何从队列中检索和删除消息。和教程一样，以下部分应按序遵照执行。

使用 AWS Management Console 或 AWS 命令行界面 (CLI) 您可以不编写代码而直接浏览 Amazon SQS。您也可以使用下列语言之一进行编码：Java、C#、或 PHP5。



Tip

在本文件的 HTML 版本中，您可以隐藏本指南中不适用于您所用编程语言的部分。在含有语言相关文本的页面的右上角，有一个语言选择菜单。选择您的语言以隐藏所有其他语言，或者选择“全部”，显示所有可用语言的示例。



您的选择会应用到本指南中的所有页面。

准备示例

您必须对示例文件进行配置更改才能完成本指南中的后续部分。



Tip

如果您不想编写代码，可以使用 AWS Management Console 或 [AWS Command Line Interface User Guide](#)。还可以使用 AWS Explorer（位于 [AWS Toolkit for Eclipse Getting Started Guide](#) 或 [AWS Toolkit for Visual Studio User Guide](#) 中），以在无需编写代码的情况下更好地了解 Amazon SQS。

AWS Management Console

无需编写代码直接浏览 Amazon SQS 的最简单的方式是利用 AWS Management Console。

使用 AWS Management Console 访问 Amazon SQS

- 登录 AWS 管理控制台，并通过以下网址打开 Amazon SQS 控制台：
<https://console.aws.amazon.com/sqs/>。

Java

请注意，Java 示例在一次调用中执行以下所有操作：创建队列、确认队列是否存在、发送消息、接收消息以及删除消息。

如需准备示例文件

1. 转到 [AWS SDK for Java](#) 页面并下载 SDK。
2. 下载 `aws-java-sdk-1.0.0.20100226.zip` 文件，然后将其解压到您计算机上指定为 `<sqs home>` 的目录。
3. 打开 AWS 证书文件：`<sqs home>/aws-java-sdk/samples/AmazonSimpleQueueService/AwsCredentials.properties`。
4. 输入您的访问密钥 ID 和私有访问密钥：

```
# Fill in your AWS Access Key ID and Secret Access Key # http://aws.amazon.com/security-credentials accessKey = <Your AWS Access Key ID>
secretKey = <Your Secret Access Key>
```

5. 保存文件。

C#

请注意，C# 示例在一次调用中执行以下所有操作：创建队列、确认队列是否存在、发送消息、接收消息以及删除消息。

如需准备示例文件

1. 转到 [AWS SDK for .NET](#) 页面并下载 SDK。
2. 运行下载的安装程序 `AWSToolsAndSDKForNet.msi`。
默认情况下安装到 `C:\Program Files\AWS SDK for .NET`。
3. 转到 `C:\Program Files\AWS SDK for .NET\Samples\AmazonSQS_Sample` 并打开下列示例解决方案之一：
 - `AmazonSQS_Sample.VS2008.sln`
 - `AmazonSQS_Sample.VS2010.sln`
4. 打开 `App.config` 文件。
5. 输入您的访问密钥 ID 和私有访问密钥：

```
<configuration> <appSettings> <add key="AWSAccessKey" value="<Your Access Key ID>" /> <add key="AWSSecretKey" value="<Your Secret Access Key>" /> </appSettings> </configuration>
```

6. 保存文件。

PHP 5

执行下列步骤前，您必须先完成[开始设置 \(p. 6\)](#)的 PHP 部分中的任务。

请注意，PHP 示例在一个 php 文件中执行以下所有操作：创建队列、列出队列、发送消息、接收消息以及删除消息。

如需准备示例文件

1. 转到 AWS 提供的 [PHP 示例代码](#)所在的位置，然后下载示例代码包。
2. 将示例代码 `SimpleQueueServiceSample.php` 提取到您计算机上的某个目录。

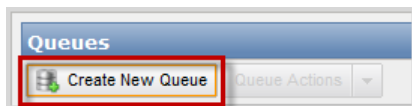
创建队列

第一个任务是使用 Amazon SQS 创建一个或多个队列。下列示例演示了如何创建名为“MyQueue”的队列。

AWS Management Console

如需运行示例

1. 登录 AWS 管理控制台，并通过以下网址打开 Amazon SQS 控制台：
<https://console.aws.amazon.com/sqs/>。
2. 单击 **新建队列**。



3. 在 **新建队列** 对话框的 **队列名称** 字段中输入 `MyQueue`，并保留其余字段的默认值设置。

Create New Queue Cancel

Please enter a name for your new queue. Queue names must be 1-80 characters in length and be composed of alphanumeric characters, hyphens (-), and underscores (_). Your queue will be created in the US West (Oregon) region.

Region: US West (Oregon)

Queue Name: MyQueue

Configure your new queue by setting queue attributes (optional).

Default Visibility Timeout: 30 seconds Value must be between 0 seconds and 12 hours.

Message Retention Period: 4 days Value must be between 1 minute and 14 days.

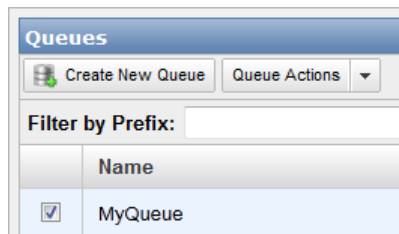
Maximum Message Size: 256 KB Value must be between 1 and 256 KB.

Delivery Delay: 0 seconds Value must be between 0 seconds and 15 minutes.

Receive Message Wait Time: 0 seconds Value must be between 0 and 20 seconds.

Cancel Create Queue

- 单击 创建队列。
您的新队列显示在队列列表中。



Java

如需运行示例

- 打开 SimpleQueueServiceSample.java。

以下部分的代码创建队列：

```
// Create a queue
System.out.println("Creating a new SQS queue called MyQueue.\n");
CreateQueueRequest createQueueRequest = new CreateQueueRe
quest().withQueueName("MyQueue");
String myQueueUrl = sqs.createQueue(createQueueRequest).getQueueUrl();
```

- 编译并运行示例。
The MyQueue queue is created.

C#

如需运行示例

1. 打开 Program.cs。

以下部分的代码创建队列：

```
//Creating a queue
Console.WriteLine("Create a queue called MyQueue.\n");
CreateQueueRequest sqsRequest = new CreateQueueRequest();
sqsRequest.QueueName = "MyQueue";
CreateQueueResponse createQueueResponse = sqs.CreateQueue(sqsRequest);
String myQueueUrl;
myQueueUrl = createQueueResponse.CreateQueueResult.QueueUrl;
```

2. 运行示例。

The MyQueue queue is created.

PHP 5

如需运行示例

1. 将示例 SimpleQueueServiceSample.php 装载到您的浏览器上，在您的 Web 服务器上运行该示例。例如，如果您使用的是 Apache Web 服务器，则可以将“SimpleQueueServiceSample.php”放在“htdocs”目录中，然后在浏览器中键入“http://localhost/SimpleQueueServiceSample.php”。
2. 结果会显示在您的浏览器窗口中。

以下部分的代码创建队列：

```
// Create the queue using the AWS-PHP-SDK function, create_queue, and store
it // in the '$newqueue' variable. $newqueue = $sqs->create_queue($queuename);
// Print the created queue results. echo/ '<br></br><h2>CREATE QUEUE</h2>';
print '<pre>' . print_r($newqueue, true) . '<pre><br/><br/>';
```

确认队列存在

创建队列时，队列可能需要较短一段时间在整个 Amazon SQS 系统中传播。您可以列出 Amazon SQS 中已有的队列，从而确认队列是否存在。下面的代码片段列出您已使用 2009 年 2 月 1 日版的 Amazon SQS 创建的队列。

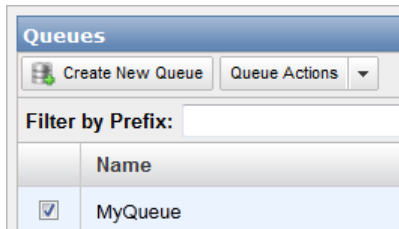
AWS Management Console

AWS Management Console 显示您在所选地区内已拥有的队列的列表。

如需显示某个特定地区的队列列表

- 从 地区 下拉列表中选择 一个地区，该列表位于控制台右上角，在“帮助”旁边。

控制台显示您在该地区的所有队列。



Java

如需运行示例

1. 打开 `SimpleQueueServiceSample.java`。

以下部分的代码列出您的队列：

```
// List queues
System.out.println("Listing all queues in your account.\n");
for (String queueUrl : sqs.listQueues().getQueueUrls()) {
    System.out.println(" QueueUrl: " + queueUrl);
}
System.out.println();
```

2. 编译并运行示例。

Amazon SQS returns the list of the queues you've created using the 2009-02-01 version of Amazon SQS, including the newly created `MyQueue` queue. Each queue is identified by its *queue URL*. The response also includes the request ID Amazon SQS assigned to your request.

C#

如需运行示例

1. 打开 `Program.cs`。

以下部分的代码列出您的队列：

```
//Confirming the queue exists
ListQueuesRequest listQueuesRequest = new ListQueuesRequest();
ListQueuesResponse listQueuesResponse = sqs.ListQueues(listQueuesRequest);

Console.WriteLine("Printing list of Amazon SQS queues.\n");
if (listQueuesResponse.IsSetListQueuesResult())
{
    ListQueuesResult listQueuesResult = listQueuesResponse.ListQueuesResult;

    foreach (String queueUrl in listQueuesResult.QueueUrl)
    {
        Console.WriteLine(" QueueUrl: {0}", queueUrl);
    }
}
```

```
}  
Console.WriteLine();
```

2. 运行示例。

Amazon SQS returns the list of the queues you've created using the 2009-02-01 version of Amazon SQS, including the newly created `MyQueue` queue. Each queue is identified by its *queue URL*. The response also includes the request ID Amazon SQS assigned to your request.

PHP 5

如需运行示例

1. 将示例 `SimpleQueueServiceSample.php` 装载到您的浏览器上，在您的 Web 服务器上运行该示例。例如，如果您使用的是 Apache Web 服务器，则可以将“`SimpleQueueServiceSample.php`”放在“`htdocs`”目录中，然后在浏览器中键入“`http://localhost/SimpleQueueServiceSample.php`”。
2. 结果会显示在您的浏览器窗口中。

以下部分的代码列出队列：

```
// Retrieve the queue list using the AWS SDK for PHP function, get_queue_list,  
and // store it in the '$queues' variable. $queues = $sqs->get_queue_list();  
// Print the formatted queue list results. echo '<h2>LIST QUEUES</h2>'; print  
'<pre>' . print_r($queues, true) . '<pre><br/><br/>'; // Retrieve the queue  
url using the AWS-PHP-SDK function, get_queue_url, which // has the  
'$queueurl' variable passed to it as a parameter. // The retrieved queue url  
for the '$queueurl' variable is then stored in the // '$queueurl' variable.  
$queueurl = $sqs->get_queue_url($queueurl); // Get the actual queue url,  
which is a CFSimpleXML object filtered from the // CFSimpleXML [body]. $url  
= (string) $queueurl->body->GetQueueUrlResult->QueueUrl; // Print the queue  
url echo '<h2>LIST QUEUE URL</h2>'; print "<pre>{$url}<pre><br/><br/>";
```

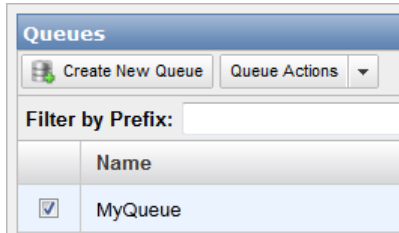
发送消息

您已确认队列在 Amazon SQS 系统中存在，可以向队列发送消息。以下代码片段演示了如何将消息 `This is my message text.` 发送到您的 `MyQueue` 队列。

AWS Management Console

如需发送消息

1. 在 AWS Management Console 中选择一个队列。

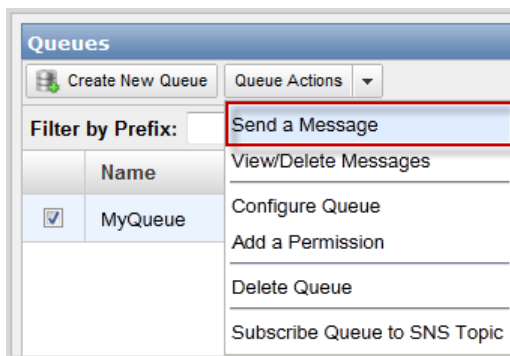


2. 从队列操作下拉列表中选择 发送消息。

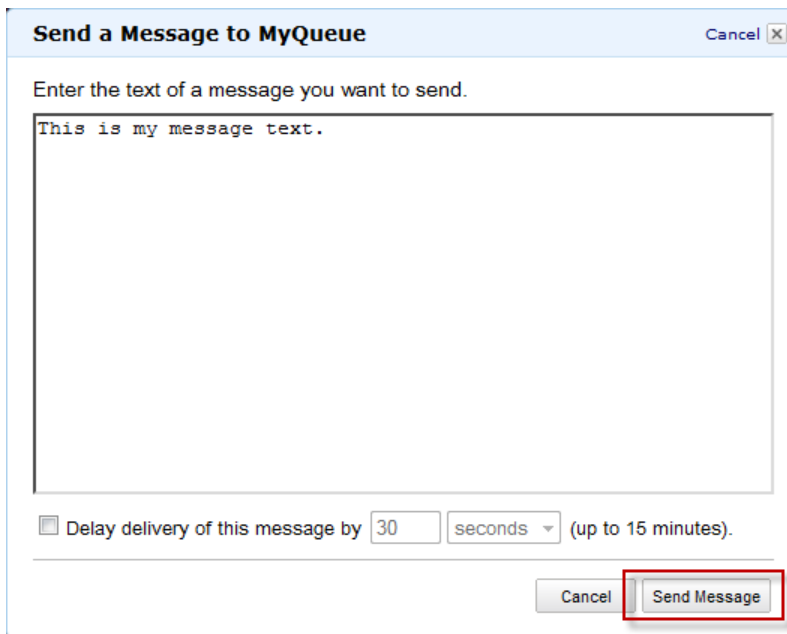


Note

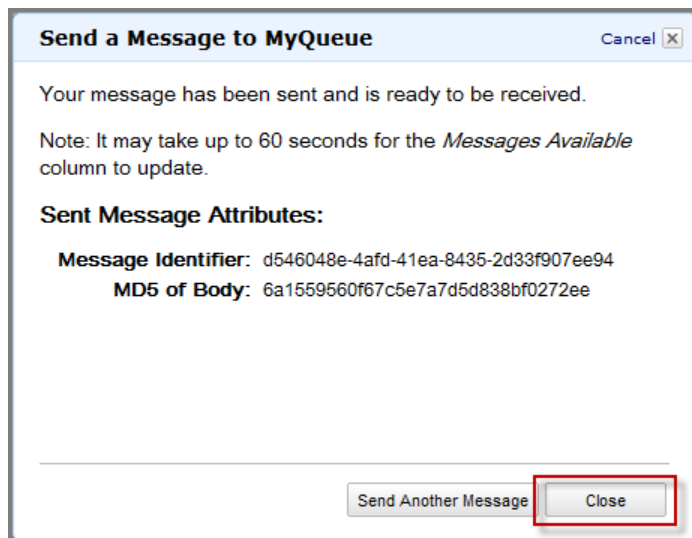
仅在已选中队列的情况下，队列操作 下拉列表才可用。



3. 在 发送消息至 MyQueue 对话框中，输入 This is my message text.，然后单击 发送消息。



4. 在 发送消息至 MyQueue 确认框中，单击 关闭。



Java

如需运行示例

1. 打开 `SimpleQueueServiceSample.java`。

以下部分的代码向您的队列发送消息：

```
// Send a message
System.out.println("Sending a message to MyQueue.\n");
sqs.sendMessage(new SendMessageRequest()
    .withQueueUrl(myQueueUrl)
    .withMessageBody("This is my message text."));
```

2. 编译并运行示例。

The message `This is my message text.` is sent to the queue. The response includes the following items:

- The *message ID* Amazon SQS assigns to the message
- An MD5 digest of the message body, which you can use to confirm that SQS received the message correctly (for information about MD5, go to <http://faqs.org/rfcs/rfc1321.html>)
- The request ID that Amazon SQS assigned to your request

C#

如需运行示例

1. 打开 `Program.cs`。

以下部分的代码向您的队列发送消息：

```
//Sending a message
Console.WriteLine("Sending a message to MyQueue.\n");
SendMessageRequest sendMessageRequest = new SendMessageRequest();
sendMessageRequest.QueueUrl = myQueueUrl; //URL from initial queue creation
sendMessageRequest.MessageBody = "This is my message text.";
sqs.SendMessage(sendMessageRequest);
```

2. 运行示例。

The message `This is my message text.` is sent to the queue. The response includes the following items:

- The *message ID* Amazon SQS assigns to the message
- An MD5 digest of the message body, which you can use to confirm that SQS received the message correctly (for information about MD5, go to <http://faqs.org/rfcs/rfc1321.html>)
- The request ID that Amazon SQS assigned to your request

PHP 5

如需运行示例

1. 将示例 `SimpleQueueServiceSample.php` 装载到您的浏览器上，在您的 Web 服务器上运行该示例。例如，如果您使用的是 Apache Web 服务器，则可以将“`SimpleQueueServiceSample.php`”放在“`htdocs`”目录中，然后在浏览器中键入“`http://localhost/SimpleQueueServiceSample.php`”。
2. 结果会显示在您的浏览器窗口中。

以下部分的代码向队列发送消息：

```
// Send a message to the queue using the AWS SDK for PHP function, send_message. $sndmessage = $sqs->send_message($url, 'HAVE A NICE DAY'); // The message body is an array element in the CFResponse Object [header]. $message = $sndmessage->header['x-aws-body']; // Print the sent message. echo '<h2>SEND MESSAGE</h2>'; print '<pre>' . print_r($message, true) . '</pre><br/><br/>';
```

接收消息

队列中现已有消息，您可以接受该消息（从队列中检索消息）。请求从队列中获取消息时，您不能指定要获取哪条消息。您只需指定希望获取的最大消息数量（最多 10 条），Amazon SQS 就会返回最大数量的消息。由于 Amazon SQS 是分布式系统并且我们在此使用的特定队列中几乎没有消息，对接收请求的响应可能为空。因此，在默认使用短轮询的本例中，您应该重新运行示例直至收到消息。



Note

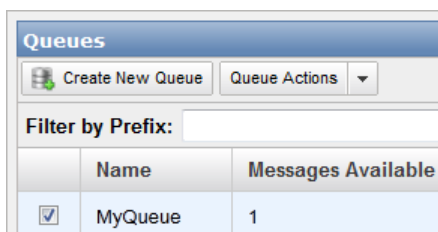
设计自己的应用程序时，您可以决定是长轮询还是短轮询对于您的环境和应用程序需求效果更好。有关更多信息，请参阅 [Amazon SQS 长轮询](#)

Amazon SQS 不会在将消息返回给您之后自动将其删除，以防您实际未收到消息（接收组件可能失效或失去连接）。您必须另行发送请求以删除消息，该请求确认您已成功接收并处理消息。有关更多信息，请参阅 [删除消息](#) (p. 25)。

AWS Management Console

如需接收消息

1. 在 AWS Management Console 中选择一个队列。

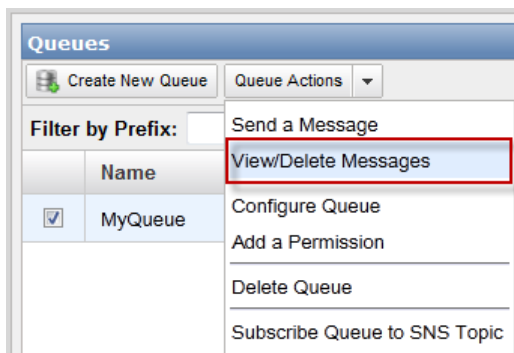


2. 从队列操作下拉列表中选择 查看/删除消息。



Note

仅在已选中队列的情况下，队列操作 下拉列表才可用。



3. 单击 开始轮询消息 以接收来自队列的消息。

View messages currently available in the queue by clicking the *Start Polling for Messages* button.

- Messages will come from the front of the queue unless other applications are also reading from the queue.
- Messages displayed in the console will not be available to other applications until the console stops polling for messages.
- The console will stop polling for messages as soon as the specified number of seconds have elapsed, the requested number of messages have been received, or the *Stop Now* button has been pressed.
- Deleting a message from the console permanently removes it from the queue.

Don't show this again. **Start Polling for Messages** ▶



Note

如果您此前没有选中 不再显示 复选框，那么 开始轮询消息 对话框将不会显示。

在 MyQueue 中查看/删除消息 对话框显示来自队列的消息。

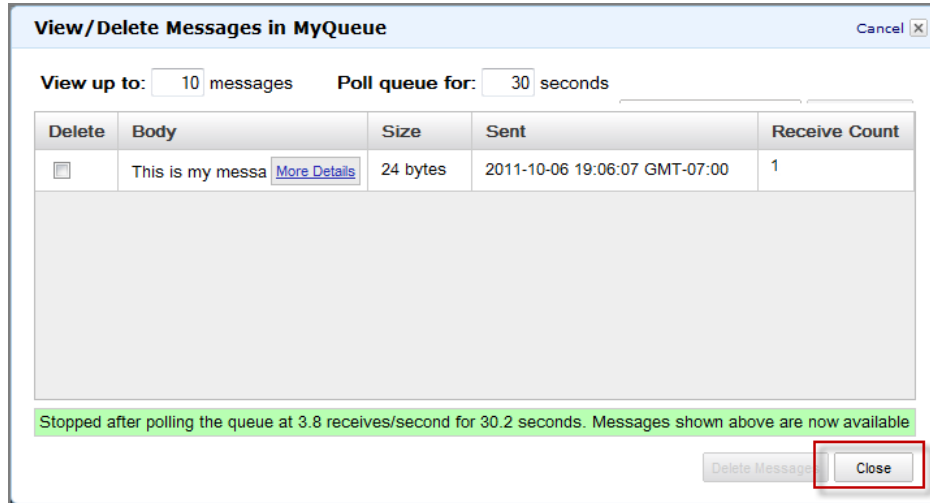
View/Delete Messages in MyQueue Cancel X

View up to: messages Poll queue for: seconds

Delete	Body	Size	Sent	Receive Count
<input type="checkbox"/>	This is my messa More Details	24 bytes	2011-10-06 19:06:07 GMT-07:00	1

Polling the queue at 3.8 receives/second. Stopping in 27.1 seconds. Messages shown above are currently hidden

对话框底部的黄色进程条显示消息可见性超时的状态。进程条为黄色时，消息对其他使用者不可见。消息转为绿色时，可见性超时结束，消息对其他使用者重新可见。



4. 单击 关闭 关闭 在 MyQueue 中查看/删除消息 对话框。

Java

如需运行示例

1. 打开 SimpleQueueServiceSample.java。

以下部分的代码从您的队列接收消息：

```
System.out.println("Receiving messages from MyQueue.\n");
ReceiveMessageRequest receiveMessageRequest = new ReceiveMessageRe
quest(myQueueUrl);
List<Message> messages = sqs.receiveMessage(receiveMessageRequest).getMes
sages();
for (Message message : messages) {
    System.out.println("  Message");
    System.out.println("    MessageId:    " + message.getMessageId());
    System.out.println("    ReceiptHandle: " + message.getReceiptHandle());
    System.out.println("    MD5OfBody:    " + message.getMD5OfBody());
    System.out.println("    Body:          " + message.getBody());
    for (Entry<String, String> entry : message.getAttributes().entrySet()) {

        System.out.println("  Attribute");
        System.out.println("    Name:    " + entry.getKey());
        System.out.println("    Value:  " + entry.getValue());
    }
}
System.out.println();
```

2. 编译并运行示例。

The MyQueue queue is polled for messages and returns 0 or more messages. The sample prints the following items:

- The *message ID* that you received when you sent the message to the queue
- The *receipt handle* (which you use later to delete the message)
- An MD5 digest of the message body (for information about MD5, go to <http://faqs.org/rfcs/rfc1321.html>)

- The message body
- The request ID that Amazon SQS assigned to your request

If no messages are received in this particular call, the response includes only the request ID.

C#

如需运行示例

1. 打开 Program.cs。

以下部分的代码从您的队列接收消息：

```
//Receiving a message
ReceiveMessageRequest receiveMessageRequest = new ReceiveMessageRequest();
receiveMessageRequest.QueueUrl = myQueueUrl;
ReceiveMessageResponse receiveMessageResponse = sqs.ReceiveMessage(receiveMessageRequest);
if (receiveMessageResponse.IsSetReceiveMessageResult())
{
    Console.WriteLine("Printing received message.\n");
    ReceiveMessageResult receiveMessageResult = receiveMessageResponse.ReceiveMessageResult;
    foreach (Message message in receiveMessageResult.Message)
    {
        Console.WriteLine("  Message");
        if (message.IsSetMessageId())
        {
            Console.WriteLine("    MessageId: {0}", message.MessageId);
        }
        if (message.IsSetReceiptHandle())
        {
            Console.WriteLine("    ReceiptHandle: {0}", message.ReceiptHandle);
        }
        if (message.IsSetMD5OfBody())
        {
            Console.WriteLine("    MD5OfBody: {0}", message.MD5OfBody);
        }
        if (message.IsSetBody())
        {
            Console.WriteLine("    Body: {0}", message.Body);
        }
        foreach (Amazon.SQS.Model.Attribute attribute in message.Attribute)
        {
            Console.WriteLine("  Attribute");
            if (attribute.IsSetName())
            {
                Console.WriteLine("    Name: {0}", attribute.Name);
            }
            if (attribute.IsSetValue())
            {
                Console.WriteLine("    Value: {0}", attribute.Value);
            }
        }
    }
}
```

```
String messageReceiptHandle = receiveMessageResponse.ReceiveMessageResult.Message[0].ReceiptHandle;
```

2. 运行示例。

The `MyQueue` queue is polled for messages and returns 0 or more messages. The sample prints the following items:

- The *message ID* that you received when you sent the message to the queue
- The *receipt handle* (which you use later to delete the message)
- An MD5 digest of the message body (for information about MD5, go to <http://faqs.org/rfcs/rfc1321.html>)
- The message body
- The request ID that Amazon SQS assigned to your request

If no messages are received in this particular call, the response includes only the request ID.

PHP 5

如需运行示例

1. 将示例 `SimpleQueueServiceSample.php` 装载到您的浏览器上，在您的 Web 服务器上运行该示例。例如，如果您使用的是 Apache Web 服务器，则可以将“`SimpleQueueServiceSample.php`”放在“`htdocs`”目录中，然后在浏览器中键入“`http://localhost/SimpleQueueServiceSample.php`”。
2. 结果会显示在您的浏览器窗口中。

以下部分的代码从队列接收消息：

```
// Receive a message from the queue using the AWS SDK for PHP function, receive_message. $received = $sqs->receive_message($url); // Print the received message results. echo '<h2>RECEIVE MESSAGE</h2>'; print '<pre>' . print_r($received, true) . '<pre><br/><br/>'; // Store the receipt handle, which is a CFSimpleXML object filtered from the // CFSimpleXML [body]. $receiptHandle = $received->body->ReceiveMessageResult->Message->ReceiptHandle; // Print receipt handle. echo '<h2>ReceiptHandle</h2>'; echo '<pre>' . print_r($receiptHandle, true) . '<pre><br/><br/>';
```

删除消息

收到消息后，必须将其从队列删除，表明您已处理且不再需要该消息。您通过提供收到消息时 Amazon SQS 返回的接收句柄，指定要删除哪条消息。您每次调用只能删除一条消息。您可以通过调用 `DeleteQueue` 删除整个队列，即使队列中含有消息。



Note

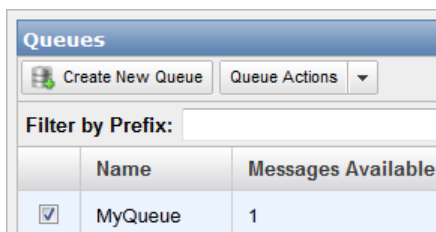
如果您没有该消息的接收句柄，可以再次调用 `ReceiveMessage` 并再次接收消息。您每次收到消息时都会获得不同的接收句柄。请在调用 `DeleteMessage` 时使用最新的接收句柄，否则您的消息可能无法从队列中删除。

下面的示例演示了如何从您的 `MyQueue` 队列删除消息。

AWS Management Console

如何删除消息

1. 在 AWS Management Console 中选择一个队列。

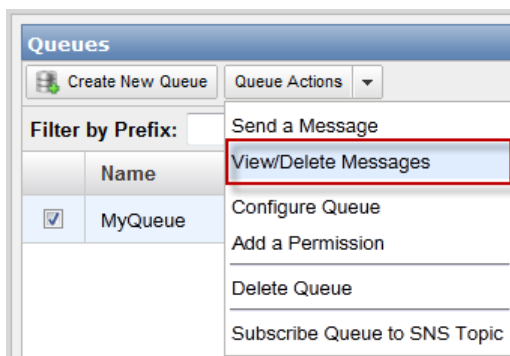


2. 从队列操作 下拉列表中选择 查看/删除消息。



Note

仅在已选中队列的情况下，队列操作 下拉列表才可用。



3. 单击 开始轮询消息，从队列中查看消息。

View messages currently available in the queue by clicking the *Start Polling for Messages* button.

- Messages will come from the front of the queue unless other applications are also reading from the queue.
- Messages displayed in the console will not be available to other applications until the console stops polling for messages.
- The console will stop polling for messages as soon as the specified number of seconds have elapsed, the requested number of messages have been received, or the *Stop Now* button has been pressed.
- Deleting a message from the console permanently removes it from the queue.

Don't show this again. **Start Polling for Messages** ▶



Note

如果您此前没有选中 不再显示 复选框，那么 开始轮询消息 对话框将不会显示。

查看/删除消息 对话框从队列中显示一条消息。

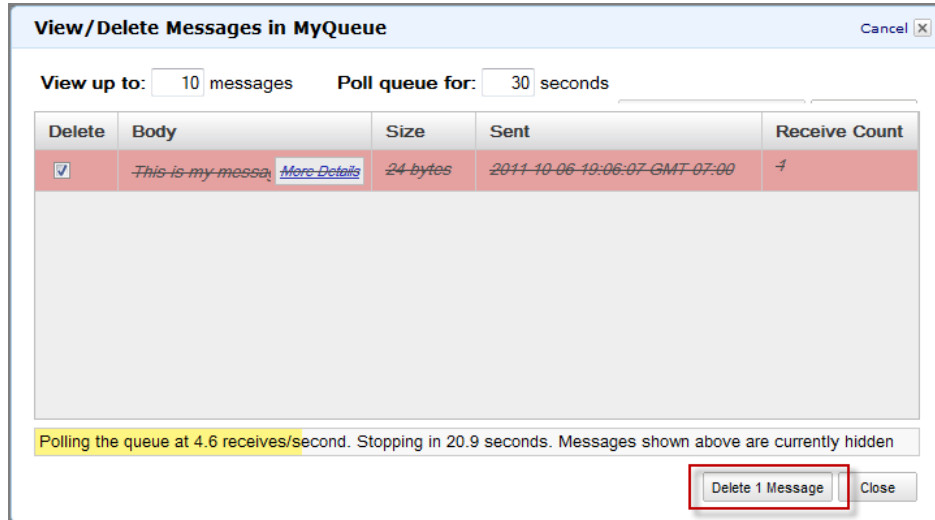
View/Delete Messages in MyQueue Cancel X

View up to: messages Poll queue for: seconds

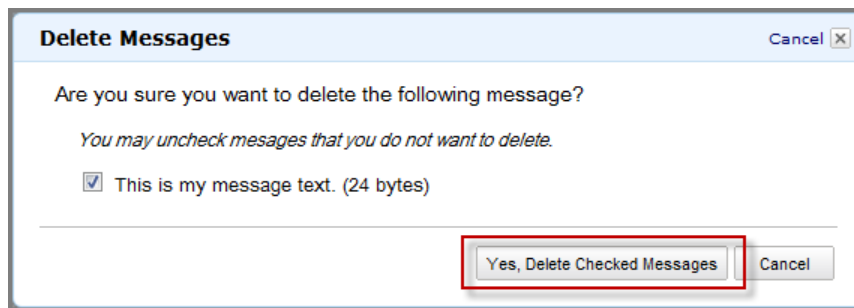
Delete	Body	Size	Sent	Receive Count
<input type="checkbox"/>	This is my messa More Details	24 bytes	2011-10-06 19:06:07 GMT-07:00	1

Polling the queue at 3.8 receives/second. Stopping in 27.1 seconds. Messages shown above are currently hidden

4. 选定您要删除的消息。



- 单击 删除 1 条消息，删除选定的消息。
此时将显示 删除消息 确认对话框。



- 单击 是，删除所选消息。
选定的消息已删除。
- 单击 关闭 以关闭 查看/删除消息 对话框。

Java

如需运行示例

- 打开 SimpleQueueServiceSample.java。

以下部分的代码删除消息：

```
// Delete a message
System.out.println("Deleting a message.\n");
String messageReceiptHandle = messages.get(0).getReceiptHandle();
sqs.deleteMessage(new DeleteMessageRequest()
    .withQueueUrl(myQueueUrl)
    .withReceiptHandle(messageReceiptHandle));
```

- 编译并运行示例。

The message is deleted from the `MyQueue` queue. The response includes the request ID that Amazon SQS assigned to your request.

C#

如需运行示例

1. 打开 `Program.cs`。

以下部分的代码删除消息：

```
//Deleting a message
Console.WriteLine("Deleting the message.\n");
DeleteMessageRequest deleteRequest = new DeleteMessageRequest();
deleteRequest.QueueUrl = myQueueUrl;
deleteRequest.ReceiptHandle = messageReceiptHandle;
sqs.DeleteMessage(deleteRequest);
```

2. 运行示例。

The message is deleted from the `MyQueue` queue. The response includes the request ID that Amazon SQS assigned to your request.

PHP 5

如需运行示例

1. 将示例 `SimpleQueueServiceSample.php` 装载到您的浏览器上，在您的 Web 服务器上运行该示例。例如，如果您使用的是 Apache Web 服务器，则可以将“`SimpleQueueServiceSample.php`”放在“`htdocs`”目录中，然后在浏览器中键入“`http://localhost/SimpleQueueServiceSample.php`”。
2. 结果会显示在您的浏览器窗口中。

以下部分的代码从队列删除消息：

```
// Deletes a message from a specific queue using the receipt handle $deletemessage = $sqs->delete_message($url, $receipthandle); // Print the delete message results. echo '<h2>DELETE MESSAGE</h2>'; echo '<pre>' . print_r($deletemessage, true) . '<pre>';
```

您已完成！

恭喜您！您已成功创建队列并发送、接受和删除消息。有关本指南中未涵盖的知识点以及如何继续的信息，请参阅[我从这里可以继续进行哪些内容？](#) (p. 29)。

我从这里可以继续进行哪些内容？

您已通读本指南，很好地了解了要使用 Amazon SQS 需要执行的主要任务，以及可以在 Amazon SQS 开发人员指南的哪些部分获取更多信息和指导。本部分描述我们建议您采取的后续步骤。

查阅文章和教程

建议您阅读[文章与教程](#)中提供的 Amazon SQS 内容。特别是下面两个专门制作的网络广播，可以帮助您更深入地了解 Amazon SQS：

- [Amazon SQS:作为粘合剂的队列](#)是一段简短的介绍，展示了 Amazon SQS 如何在服务导向型架构 (SOA) 应用程序中与其他元素集成。
- [使用 Amazon SQS 建立自我修复式应用程序](#)介绍了将 Amazon SQS 用作工作流队列开发应用程序的概念。这些理念有助于确保工作流中的所有进程得以完成。

阅读论坛

我们建议您看一看 [Amazon SQS 论坛](#)，了结其他用户在做什么及其已提出的问题。这样做有助于您进一步理解您可以利用本服务做什么以及不能做什么。

查看其他可用示例代码

您已了解了随本指南提供的示例代码（有关更多信息，请参阅[准备示例 \(p. 10\)](#)）。您可以查看[示例代码和库](#)页面上提供的其他 Amazon SQS 示例代码，也可以访问针对具体的编程语言的开发人员中心：

- [Java 开发人员中心](#)
- [PHP 开发人员中心](#)
- [Python](#)
- [Ruby 开发人员中心](#)
- [Windows 和 .NET 开发人员中心](#)

AWS 账户和安全证书

到现在为止，您已经注册了服务，获得了 AWS 账户和安全证书，之后又完成了涵盖产品关键功能的一个小练习。既然您已经完成了练习，我们建议您咨询组织内部的管理员或同事，确定他/她是否已为您准备好 AWS 账户和安全证书，以便您将来用于与 AWS 进行交互。

如果您是账户所有者或管理员，并且想详细了解 AWS Identity and Access Management 的相关信息，请访问 <http://aws.amazon.com/iam> 阅读产品介绍，或者参阅 [Using AWS Identity and Access Management](#) 中的技术文档。

文档历史记录

下表描述了自上次发行 *Amazon Simple Queue Service 入门指南* 以来对文档所做的重要更改。

- API 版本：2012-11-05
- 最近文档更新时间：2012 年 11 月 21 日

变更	描述	修改日期
新功能	Amazon SQS 现已通过 AWS Management Console 提供。有关如何借助 AWS Management Console 创建队列、发送消息、接收消息的示例，请参阅 使用 Amazon SQS (p. 10) 。	2012 年 11 月 21 日
新功能	该服务现已集成 AWS Identity and Access Management (IAM)。有关更多信息，请参阅 使用 IAM 中的支持 IAM 的 AWS 服务部分 。	2010 年 9 月 2 日
示例代码	将 Java 示例代码替换成了 AWS SDK for Java 中的新的 Java 示例代码。有关更多信息，请参阅 使用 Amazon SQS (p. 10) 中列出的 Java 示例。	2010 年 3 月 22 日
示例代码	将 C# 和 .NET 示例代码替换成了 AWS SDK for .NET 中的新的 C# 示例代码。有关更多信息，请参阅 使用 Amazon SQS (p. 10) 中列出的 C# 示例。	2009 年 11 月 11 日
更新	更新后的指南反映了 AWS 为 Java、C#、Perl、PHP5 和 VB.NET 所提供示例代码的变化。	2008 年 12 月 15 日