Storing objects with Amazon S3 and .NET

# Overview

Amazon S3 is object storage built to store and retrieve any amount of data from anywhere – web sites and mobile apps, corporate applications, and data from IoT sensors or devices. It is designed to deliver 99.999999999% durability, and stores data for millions of applications used by market leaders in every industry. S3 provides comprehensive security and compliance capabilities that meet even the most stringent regulatory requirements. It gives customers flexibility in the way they manage data for cost optimization, access control, and compliance.

For this walk-through, we’ll create a new .NET Core console application and add the AWS SDK for S3 to it and walk through common functionality such as creating buckets, writing/reading objects, listing buckets and objects, and deleting buckets and objects.

# Prerequisites

* .NET Core 2.0 or higher installed
* AWS Account with credentials configured locally in Visual Studio or using the CLI
* Optional: Visual Studio 2017 (you can also use the command line for .NET Core)

# Create and Configure a Console Application

**Step 1: Create an empty console application project**

We’ll create a new directory, and then create the console app project in it using the following commands in either a Windows command prompt, Mac OS X bash shell or Linux bash shell:

1. mkdir s3-sample
2. cd s3-sample
3. dotnet new console

The *dotnet new* command will create the project files, and restore packages referenced by the template.

**Step 2: Add the S3 NuGet Package**

**Command Line:**

Add the NuGet package *AWSSDK.S3* to the project with the following command in either a Windows command prompt or Mac OS X or Linux bash shell:

1. dotnet add package AWSSDK.S3

**Step 3: Edit the C# Code**

You can download the complete project here and then edit the *Program.cs* file or copy/paste the code here:

|  |
| --- |
| using System; using System.IO; using System.Text; using System.Threading.Tasks; using Amazon; using Amazon.S3; using Amazon.S3.Model;  namespace S3\_Example {     class Program     {         static string bucketName = "bucket-" + Guid.NewGuid().ToString("n").Substring(0, 8);         static string key = "key-" + Guid.NewGuid().ToString("n").Substring(0, 8);          static void Main(string[] args)         {             AmazonS3Client s3 = new AmazonS3Client(RegionEndpoint.USWest2);              CreateBucket(s3);             Console.WriteLine("Press enter to continue...");             Console.Read();              ListBuckets(s3);             Console.WriteLine("Press enter to continue...");             Console.Read();              WriteObject(s3);             Console.WriteLine("Press enter to continue...");             Console.Read();              ListObjects(s3);             Console.WriteLine("Press enter to continue...");             Console.Read();              ReadObject(s3);             Console.WriteLine("Press enter to continue...");             Console.Read();              DeleteObject(s3);             Console.WriteLine("Press enter to continue...");             Console.Read();              DeleteBucket(s3);         }          static void CreateBucket(AmazonS3Client s3)         {             PutBucketRequest req = new PutBucketRequest();             req.BucketName = bucketName;             req.BucketRegion = S3Region.USW2;              Task<PutBucketResponse> res = s3.PutBucketAsync(req);              Task.WaitAll(res);              if (res.IsCompletedSuccessfully)             {                 Console.WriteLine("New S3 bucket created: {0}", bucketName);             }         }          static void WriteObject(AmazonS3Client s3)         {             MemoryStream ms = new MemoryStream(Encoding.UTF8.GetBytes("Test S3 data"));              PutObjectRequest req = new PutObjectRequest();             req.BucketName = bucketName;             req.Key = key;             req.InputStream = ms;              Task<PutObjectResponse> res = s3.PutObjectAsync(req);              Task.WaitAll(res);              if (res.IsCompletedSuccessfully)             {                 Console.WriteLine("Created object '{0}' in bucket '{1}'", key, bucketName);             }         }          static void ReadObject(AmazonS3Client s3)         {             GetObjectRequest req = new GetObjectRequest();             req.BucketName = bucketName;             req.Key = key;              Task<GetObjectResponse> res = s3.GetObjectAsync(req);              Task.WaitAll(res);              if (res.IsCompletedSuccessfully)             {                 using (TextReader tr = new StreamReader(res.Result.ResponseStream))                 {                     Console.WriteLine("Retrieved contents of object '{0}' in bucket '{1}'", key, bucketName);                     Console.WriteLine(tr.ReadToEnd());                 }             }         }          static void ListBuckets(AmazonS3Client s3)         {             ListBucketsRequest req = new ListBucketsRequest();              Task<ListBucketsResponse> res = s3.ListBucketsAsync(req);              Task.WaitAll(res);              Console.WriteLine("List of S3 Buckets in your AWS Account");             foreach (var bucket in res.Result.Buckets)             {                 Console.WriteLine(bucket.BucketName);             }         }          static void ListObjects(AmazonS3Client s3)         {             ListObjectsRequest req = new ListObjectsRequest();             req.BucketName = bucketName;             req.MaxKeys = 100;              Task<ListObjectsResponse> res = s3.ListObjectsAsync(req);              Task.WaitAll(res);              Console.WriteLine("List of objects in your S3 Bucket '{0}'", bucketName);               foreach (var s3Object in res.Result.S3Objects)             {                 Console.WriteLine(s3Object.Key);             }         }          static void DeleteObject(AmazonS3Client s3)         {             DeleteObjectRequest req = new DeleteObjectRequest();             req.BucketName = bucketName;             req.Key = key;              Task<DeleteObjectResponse> res = s3.DeleteObjectAsync(req);              Task.WaitAll(res);              if (res.IsCompletedSuccessfully)             {                 Console.WriteLine("Deleted object '{0}' from bucket '{1}'", key, bucketName);             }         }          static void DeleteBucket(AmazonS3Client s3)         {             DeleteBucketRequest req = new DeleteBucketRequest();             req.BucketName = bucketName;              Task<DeleteBucketResponse> res = s3.DeleteBucketAsync(req);              Task.WaitAll(res);              if (res.IsCompletedSuccessfully)             {                 Console.WriteLine("Deleted bucket - '{0}'", bucketName);             }         }     } } |

Figure 4 – Program.cs file

This sample code demonstrates a common pattern when developing with the AWS SDK for .NET, which is the use of a client object to represent an AWS service. That client object then exposes functionality via methods, such as the “*CreateBucket*” method in this example.

# Use S3!

To run through the S3 functionality using the console application, use the following command to build (compile) and run the app:

dotnet run

If you want to run the app again without compiling, just pass the --no-build flag like this:

dotnet run --no-build

The console application will run.