Luxoft
A DXC Technology Company

AWS Software Development Kit (SDK) for .NET

Author: Luis Coco Enríquez

Agenda

Topic
What is AWS SDK for .NET?.
Most common use cases.
How to set up your environment.
API Reference.

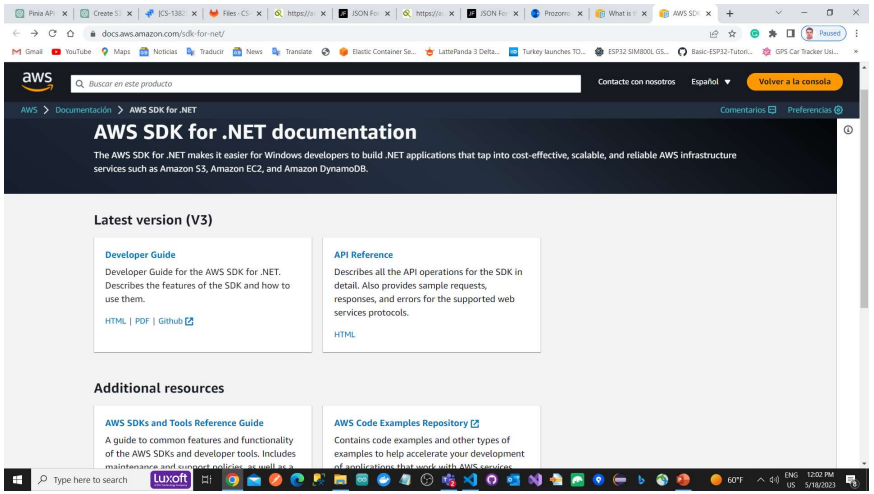
Agenda

Topic
Code examples with guidance for the AWS SDK for .NET.
Additional code examples for the AWS SDK for .NET.
GitHub repositories.
C# code samples with AWS SDK for .NET.

What is AWS SDK for .NET?

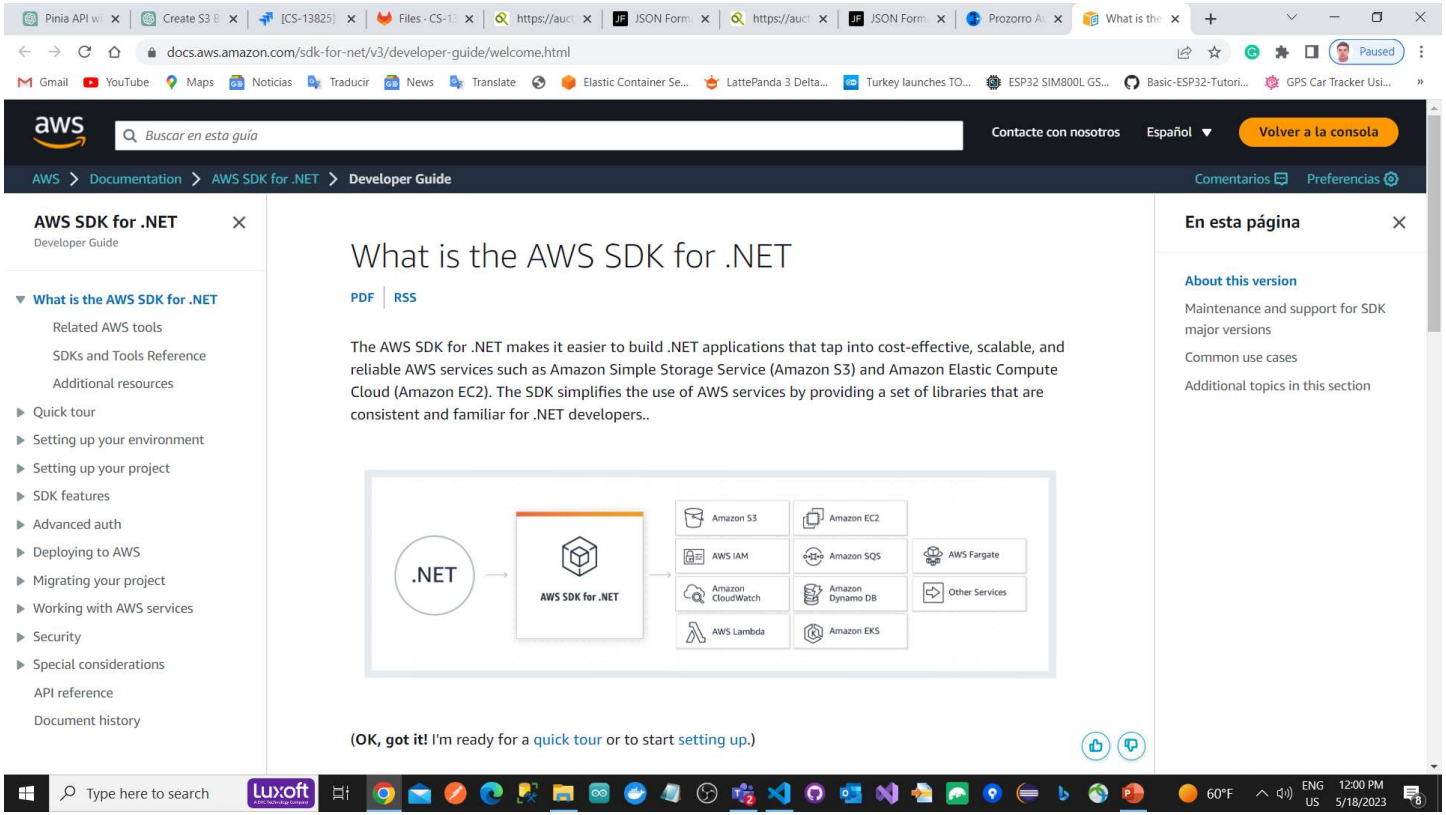
The **AWS SDK for .NET is a set of libraries (APIs)** and tools that allow .NET developers **to interact with** various **AWS services using the .NET framework**. It provides a convenient way to access AWS services such as Amazon S3, Amazon DynamoDB, Amazon EC2, and many more.

To use the AWS SDK for .NET, you need to install the **AWSSDK** packages using **NuGet**, which is the core library that provides the foundational functionality for working with AWS services. Additionally, you can install other specific service packages like **AWSSDK.S3** or **AWSSDK.DynamoDB** depending on the services you want to interact with.



https://docs.aws.amazon.com/sdk-for-net/

What is AWS SDK for .NET?



https://docs.aws.amazon.com/sdk-for-net/v3/developer-guide/welcome.html

Luxoft
A DXC Technology Company

Most common use cases

The AWS SDK for .NET provides a wide range of functionalities that enable developers to build various types of applications and services using the .NET framework. Some of the most common use cases for the AWS SDK for .NET are:

Building web applications: You can leverage the SDK to interact with various AWS services like Amazon S3 for storage, Amazon DynamoDB for NoSQL database, Amazon SES for sending emails, Amazon SQS for message queuing, and many more.

Serverless computing: The AWS SDK for .NET integrates with AWS Lambda, allowing you to build serverless applications using .NET. You can create Lambda functions, interact with other AWS services, and respond to events using the SDK.

Data processing and analytics: The SDK provides APIs to work with AWS services like Amazon Redshift for data warehousing, Amazon EMR for big data processing, Amazon Kinesis for real-time streaming data, and Amazon Athena for interactive query analysis.

Internet of Things (IoT): You can use the SDK to develop IoT applications by interacting with AWS IoT Core, which enables secure communication between devices and the cloud.

Most common use cases

Machine Learning: The AWS SDK for .NET supports AWS services like Amazon SageMaker, Amazon Rekognition, and Amazon Comprehend, allowing you to build and deploy machine learning models and perform tasks like image recognition, natural language processing, and more.

DevOps and Infrastructure Automation: The SDK integrates with AWS CloudFormation for infrastructure as code, AWS Systems Manager for managing resources, AWS CodeDeploy for deploying applications, AWS CloudWatch for monitoring and logging, and other services to automate and manage your infrastructure and deployments.

Mobile and gaming applications: The SDK provides support for AWS Mobile services, including Amazon Cognito for user authentication and authorization, Amazon Mobile Analytics for tracking app usage, Amazon Pinpoint for user engagement, and Amazon GameLift for building and scaling multiplayer games.

Integration with third-party libraries and frameworks: The AWS SDK for .NET is compatible with popular .NET libraries and frameworks, allowing you to integrate AWS services seamlessly into your existing applications or frameworks like ASP.NET, Xamarin, Unity, and more.

How to setup your environment

Follow these steps to install and run the AWS SDK for .NET:

1. Create a free AWS account. See video: How to sign up for the AWS free tier in 2023

2. Create an administrative user. See video: How to create Admin User for AWS account

3. Login with the new user and See video: How to get AWS access key and secret key

4. Install AWS CLI. See video: How to download install and configure AWS CLI on Windows

5. Run the command “aws configure” to configure your AWS account (enter your region “eu-west-3”, your default output format “json”). See video: AWS CLI Tutorial

6. (Optional) You can create different profiles and switch between them (see the following slides).

How to create and modify AWS profiles

1. Open a command line as administrator.

2. For **listing** your current **profile**: **aws configure list**
   For **listing all** your AWS **profiles** run the command: **aws configure list-profiles**

3. To **add a new profile**, first login in AWS Console and generate an access_key_id and a secret_access_key.
Then run the command: **aws configure --profile second_user**.
Enter the profile region (for example: eu-west-3) and default output format (for example: json)

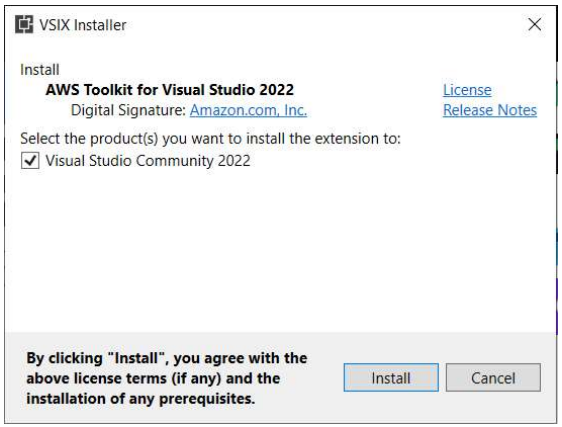
4. Profiles are stored under the "**config**" and "**credentials**" files: "C:\Users\LEnriquez**.aws**"

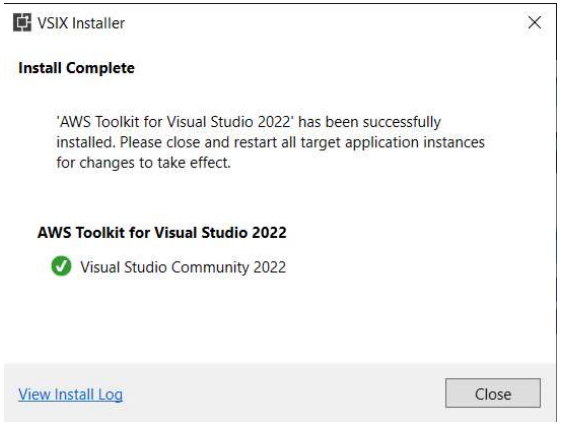
5. To switch between different profiles run the command: **set AWS_PROFILE=profile_name**
https://www.simplified.guide/aws/cli/configure-multiple-profiles

https://dev.to/hmintoh/how-to-use-multiple-aws-accounts-with-the-aws-cli-3lge#:~:text=To%20switch%20between%20different%20AWS,variable%20to%20a%20different%20value.

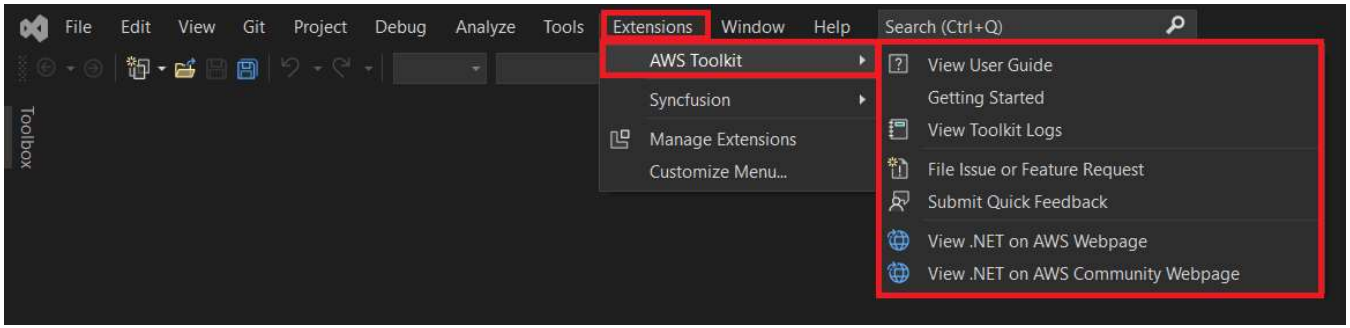
(Optional) Install AWS Toolkit for Visual Studio 2022

Donwload and install AWS Toolkit https://aws.amazon.com/es/visualstudio/



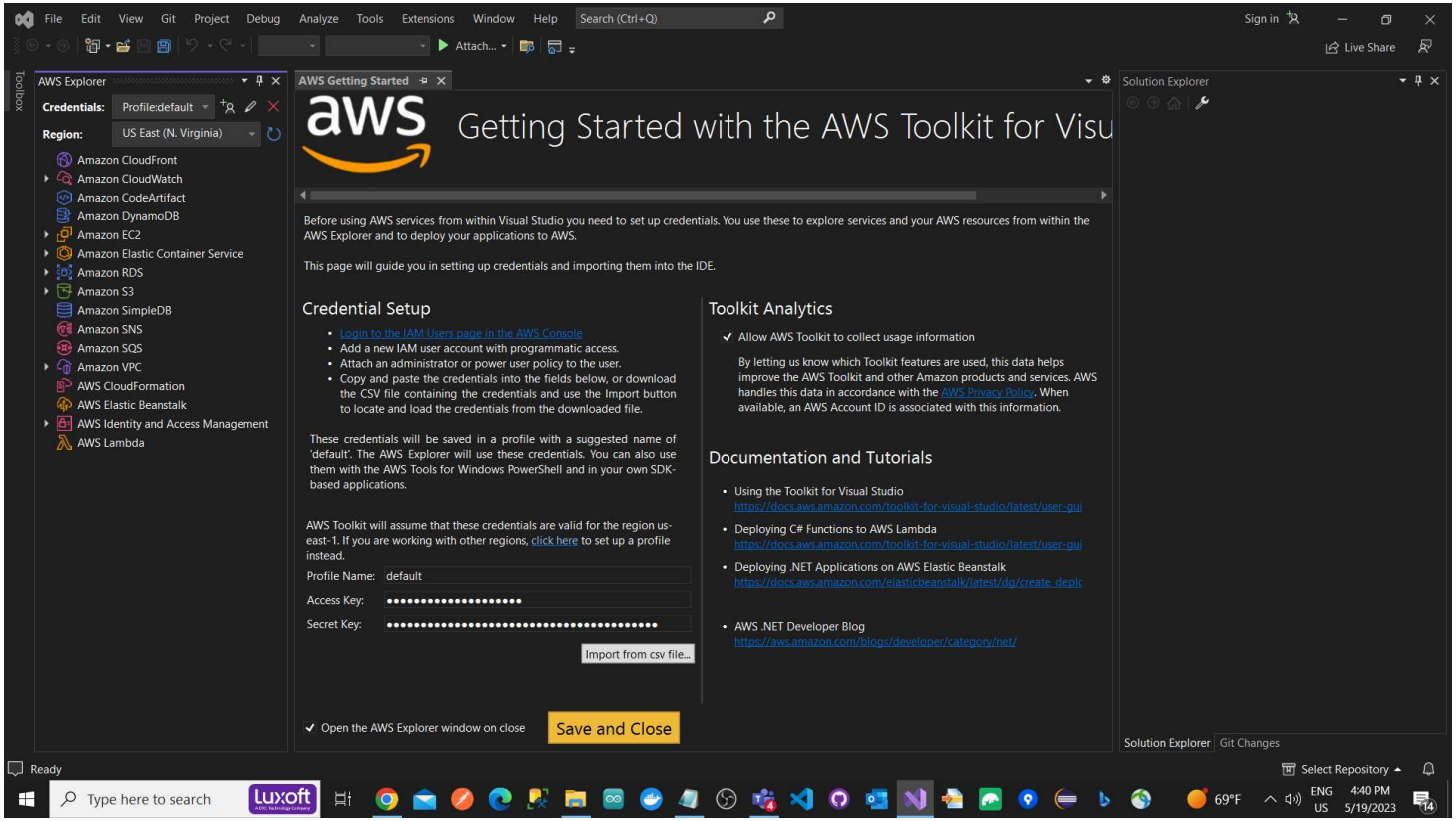


Run Visual Studio and select the menu option: Menu->**AWS Toolkit->Getting Started**

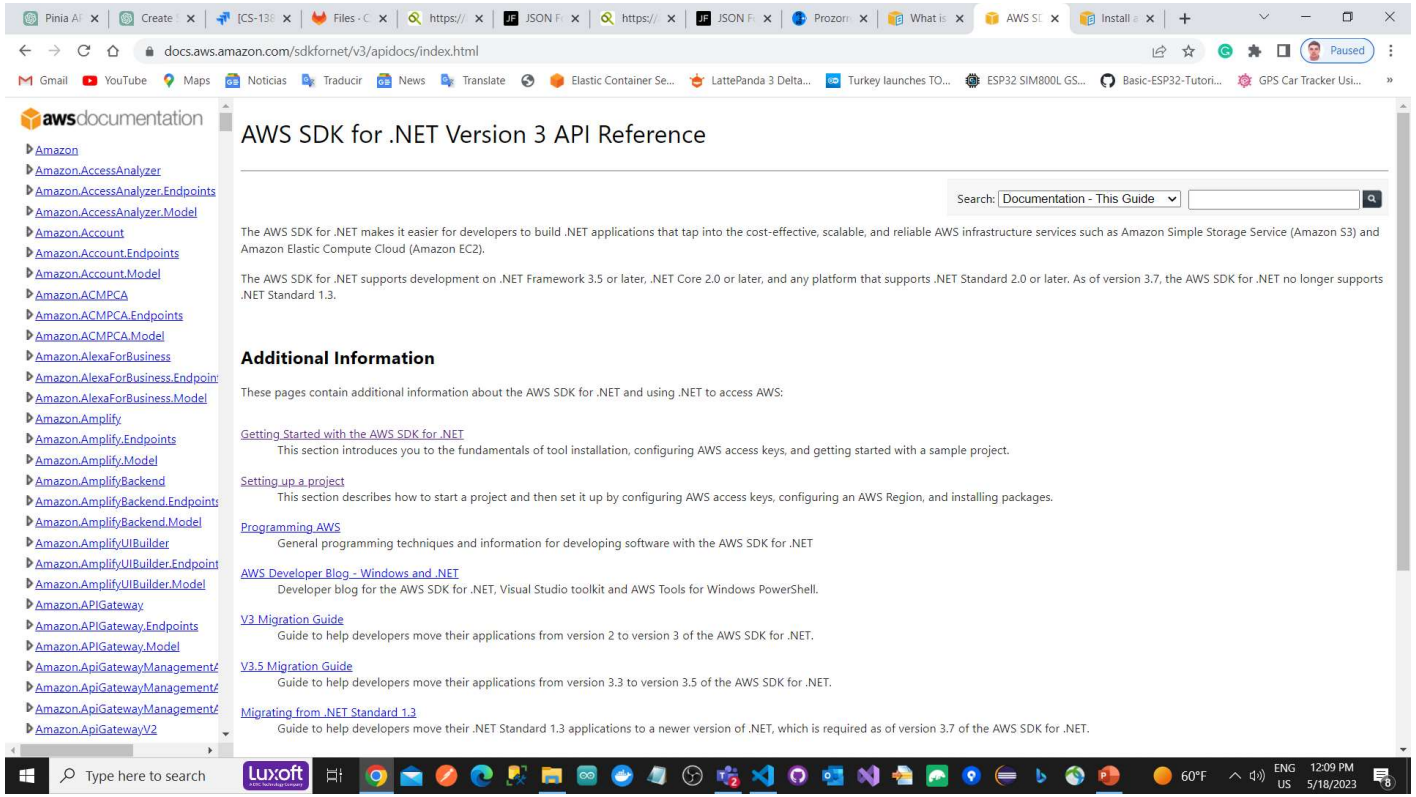


(optional) Install AWS Toolkit for Visual Studio 2022

Select in the Menu->View>**AWS Explorer**

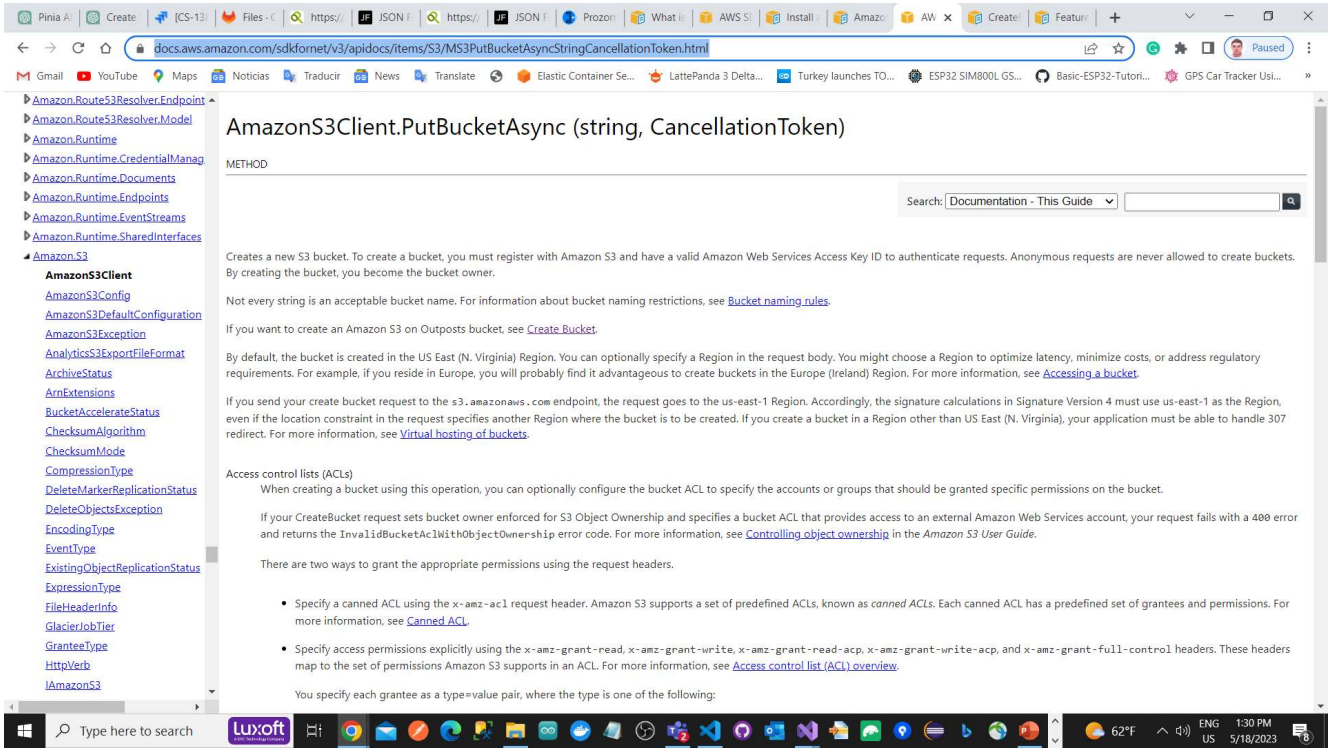


API reference



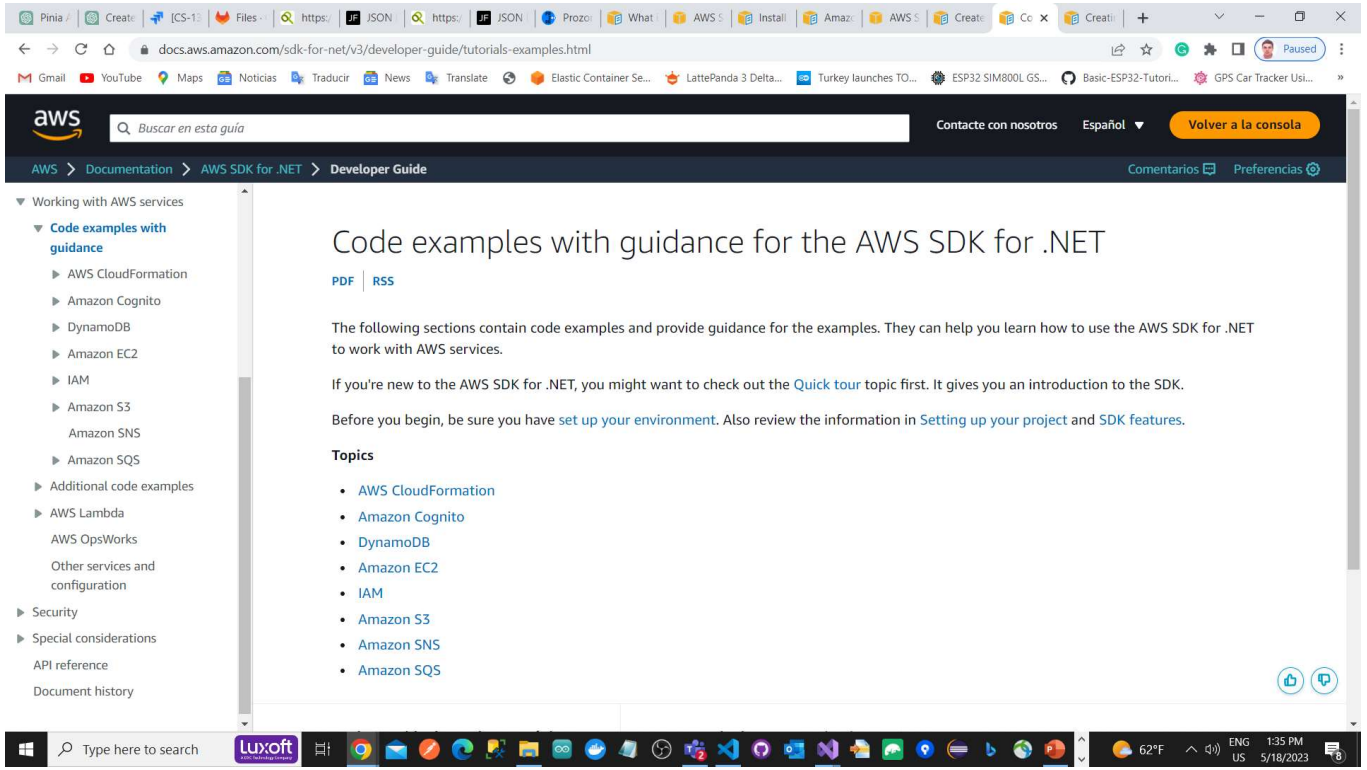
https://docs.aws.amazon.com/sdkfornet/v3/apidocs/index.html

API reference



https://docs.aws.amazon.com/sdkfornet/v3/apidocs/items/S3/MS3PutBucketAsyncStringCancellationToken.html

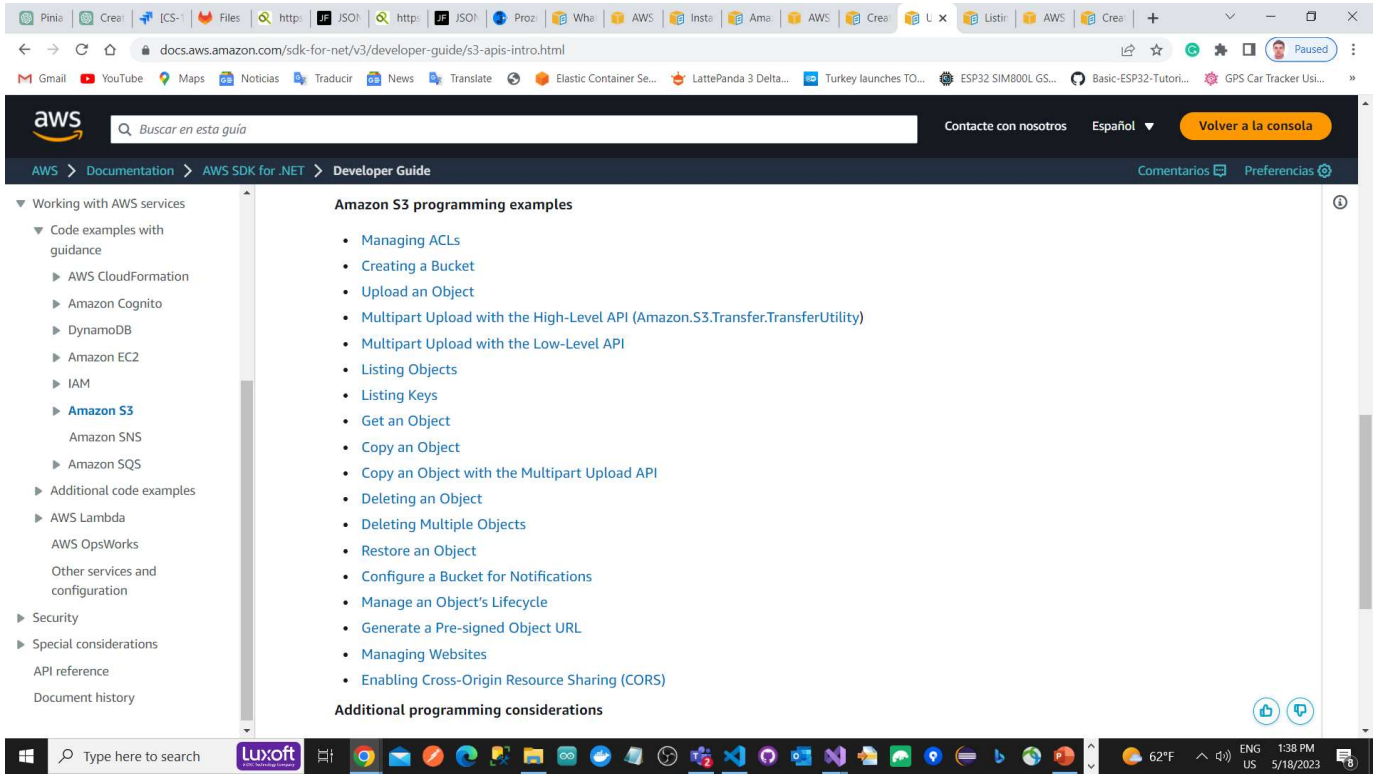
Code examples with guidance for the AWS SDK for .NET



https://docs.aws.amazon.com/sdk-for-net/v3/developer-guide/tutorials-examples.html

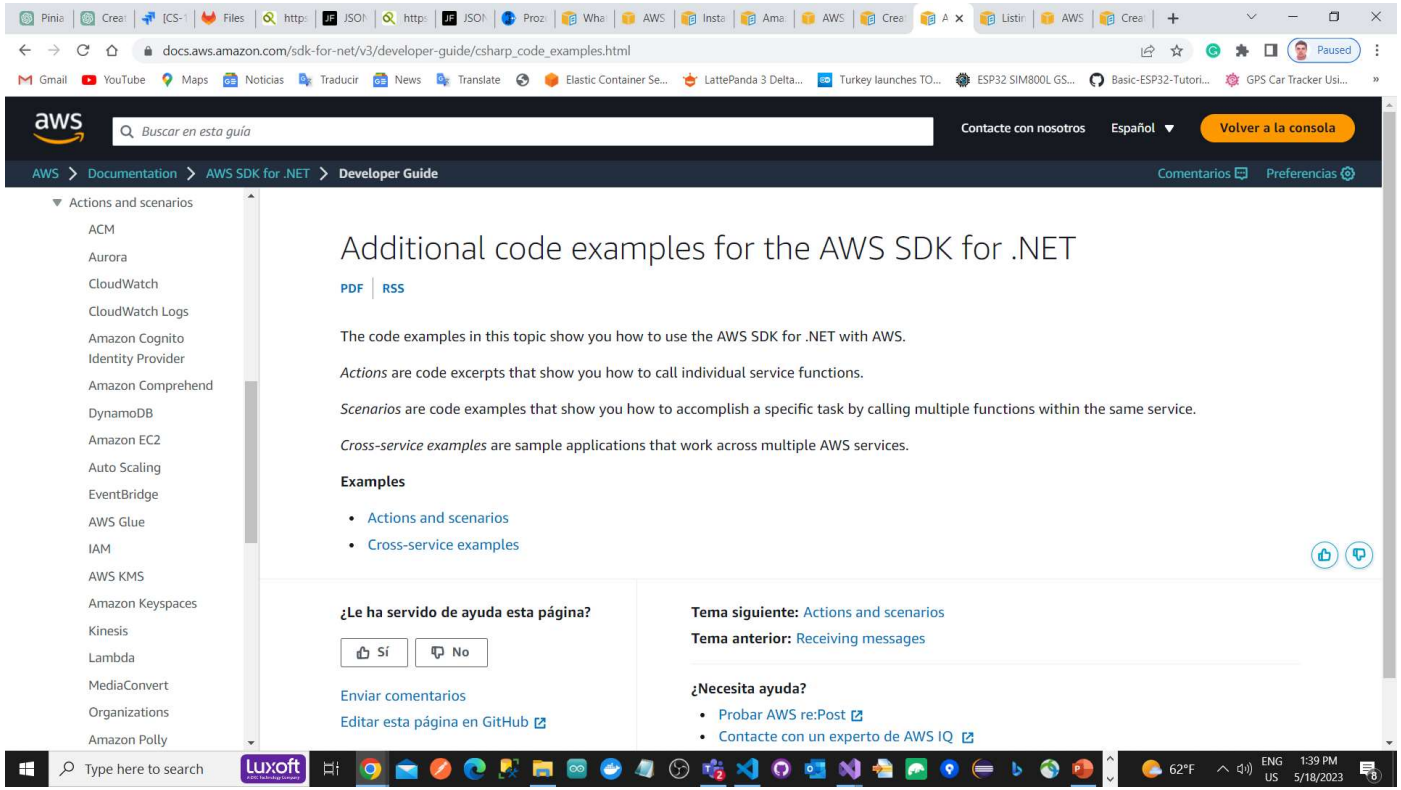
Luxoft
A DXC Technology Company

Code examples with guidance for the AWS SDK for .NET



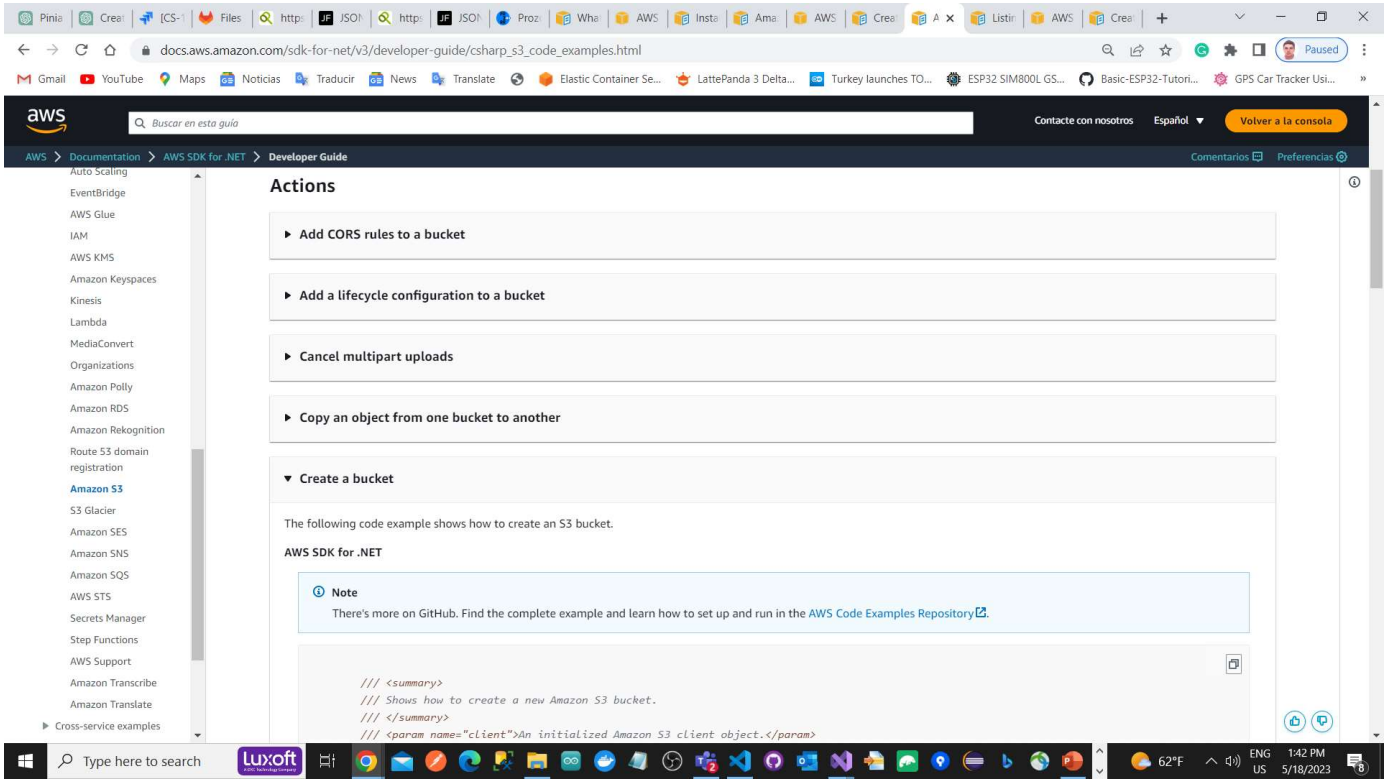
https://docs.aws.amazon.com/sdk-for-net/v3/developer-guide/s3-apis-intro.html

Additional code examples for the AWS SDK for .NET



https://docs.aws.amazon.com/sdk-for-net/v3/developer-guide/csharp_code_examples.html

Additional code examples for the AWS SDK for .NET

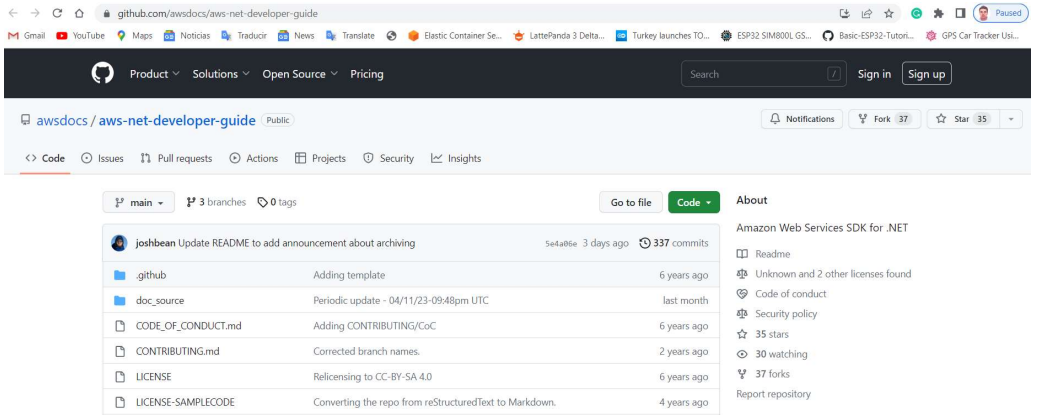


https://docs.aws.amazon.com/sdk-for-net/v3/developer-guide/csharp_s3_code_examples.html

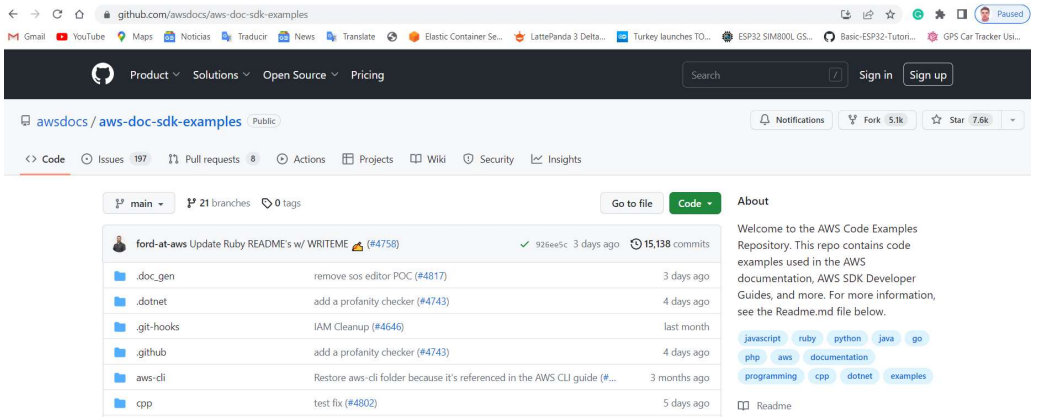
Luxoft
A DXC Technology Company

GitHub repositories

The Developer guide for the AWS SDK for .NET
https://github.com/awsdocs/aws-net-developer-guide



AWS SDK for .NET 3.x documentation examples
https://github.com/awsdocs/aws-doc-sdk-examples/tree/main/dotnetv3



Luxoft
A DXC Technology Company

C# code samples with AWS SDK for .NET

Sample 1: create a S3 bucket

https://docs.aws.amazon.com/AmazonS3/latest/userguide/example_s3_CreateBucket_section.html

Step 1: Create a new Console .NET project
Start by creating a new .NET project in your preferred IDE (e.g., Visual Studio).

Step 2: Add the necessary NuGet package: add the **AWSSDK.S3**

Step 3: Write the code
Open the code file Program.cs. Import the required namespaces at the top of the file:

```
using Amazon.S3;
using Amazon.S3.Model;

IAmazonS3 client = new AmazonS3Client();
var request = new PutBucketRequest
{
    BucketName = "luiscocoenriquezbucket",
    UseClientRegion = true,
};
var response = await client.PutBucketAsync(request);
```

C# code samples with AWS SDK for .NET

Sample 2: list S3 buckets

https://docs.aws.amazon.com/AmazonS3/latest/userguide/example_s3_ListBuckets_section.html

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
…
        var credentials = new Amazon.Runtime.BasicAWSCredentials(accessKey, secretKey);
        var config = new AmazonS3Config
        {
            RegionEndpoint = RegionEndpoint.USWest2 // Replace with your desired region
        };

        using (var client = new AmazonS3Client(credentials, config))
        {
            var request = new ListBucketsRequest();
            var response = client.ListBuckets(request);

            foreach (var bucket in response.Buckets)
            {
                Console.WriteLine(bucket.BucketName);
            }
        }
…
```

C# code samples with AWS SDK for .NET

Sample 3: upload an object to a S3 bucket

https://docs.aws.amazon.com/AmazonS3/latest/userguide/example_s3_PutObject_section.html

```
…
 var putRequest = new PutObjectRequest

{
    BucketName = bucketName,
    Key = keyName,
    ContentBody = "sample text",
    ServerSideEncryptionMethod = ServerSideEncryptionMethod.AES256,
};

var putResponse = await client.PutObjectAsync(putRequest);

// Determine the encryption state of an object.
    GetObjectMetadataRequest metadataRequest = new GetObjectMetadataRequest
    {
        BucketName = bucketName,
        Key = keyName,
    };
    GetObjectMetadataResponse response = await client.GetObjectMetadataAsync(metadataRequest);
    ServerSideEncryptionMethod objectEncryption = response.ServerSideEncryptionMethod;
    Console.WriteLine($"Encryption method used: {0}", objectEncryption.ToString());
…
```

C# code samples with AWS SDK for .NET

Sample 4: EC2 instance actions
Sample 5: Simple Queue Service (SQS)
Sample 6: DynamoDB
Sample 7: IAM Identity and Access Management services
Sample 8: SES (Simple Email Service)
Sample 9: Autoscaling Groups (EC2)
Sample 10: Aurora (RDS)
Sample 11: EventBridge
Sample 12: RDS (Relational Database Service)
Sample 13: KMS (Keys Management System)
Sample 14: Secrets Manager
Sample 15: SNS (Simple Notification Service)
Sample 16: Lambda

All these examples source code can be downloaded from the public GitHub repository:
https://github.com/luiscoco/