Storing data with Amazon DynamoDB and .NET

# Overview

Amazon DynamoDB is a fast and flexible NoSQL database service for all applications that need consistent, single-digit millisecond latency at any scale. It is a fully managed cloud database and supports both document and key-value store models. Its flexible data model, reliable performance, and automatic scaling of throughput capacity make it a great fit for mobile, web, gaming, ad tech, IoT, and many other applications

For this walk-through, we’ll create a new .NET Core Web API application and add the AWS SDK for DynamoDB and the AWS .NET Core Extensions. Each API controller group will demostrate common functionality in DynamoDB such as creating tables, writing/reading items, listing tables and items, and deleting tables and items.

# Prerequisites

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

# Included in the Web API Application

This sample code demonstrates how to use the *AWSSDK.Extensions.NETCore.Setup* package to configure and add an AWS SDK Dynamo client through the built in DI container provided by .NET Core. Additionally, each API Controller group demonstrates the three programming models available for working with Dynamo through the .NET SDK. For more information about the .NET DynamoDB programming models, [see this documentation](https://docs.aws.amazon.com/sdk-for-net/v2/developer-guide/dynamodb-intro.html).

# Use DynamoDB!

To run through the DynamoDB functionality using the Web API project, 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 Web API will run and you can use the *PowerShell* or *curl* scripts included in the Web API project README file to test the functionality.