Programming with the .NET SDK

Leonard Lobel
CTO, SLEEK TECHNOLOGIES
lennilobel.wordpress.com

Client Development

Build web-scale applications

Uses REST/HTTP

Platform SDKs

.NET / .NET Core

Java

Node.js

Python

Introducing the .NET SDK for the SQL API

Create a DocumentClient instance

Supply connection information
(endpoint and key)

Invoke methods to access resources

Create, modify, and delete resources

Use POCOs or dynamics for document objects

Task Parallel Library (TPL)

Simplified asynchronous programming

Use async/await keywords with Task objects

Introducing the .NET SDK for the SQL API

Synchronous code

```
private void Main()
{
    DoSomething();
}

private void DoSomething()
{
    // do some work
}
```

Asynchronous code

```
private async void Main()
{
    await DoSomething();
}

private async Task DoSomething()
{
    // do some asynchronous work
}
```

access resources

l delete resources

; for document objects

Task Parallel Library (TPL)

Simplified asynchronous programming

Use async/await keywords with Task objects

Introducing the .NET SDK for the SQL API

Synchronous code

```
private void Main()
{
    var result = GetSomething();
}

private string GetSomething()
{
    // do some work
    return "Hello";
}
```

Asynchronous code

```
private async void Main()
{
    var result = await GetSomething();
}

private async Task<string> GetSomething()
{
    // do some asynchronous work
    return "Hello";
}
```

access resources

l delete resources

; for document objects

Task Parallel Library (TPL)

Simplified asynchronous programming

Use async/await keywords with Task objects

Introducing the .NET SDK for the SQL API

Create a DocumentClient instance

Supply connection information (endpoint and key)

Invoke methods to access resources

Create, modify, and delete resources

Use POCOs or dynamics for document objects

Task Parallel Library (TPL)

Simplified asynchronous programming

Use async/await keywords with Task objects

LINQ provider

Automatically translates LINQ queries to SQL

Demo

Getting started with the .NET SDK

Demo

Working with databases

Demo

Working with collections

Demo

Creating documents

Demo

Querying for documents

Demo

Replacing and deleting documents

Indexing Policies

Hash index

Equality queries

Strings and numbers

Range index

Equality, range, ORDER BY

Strings and numbers

Spatial index

Distance and intersection

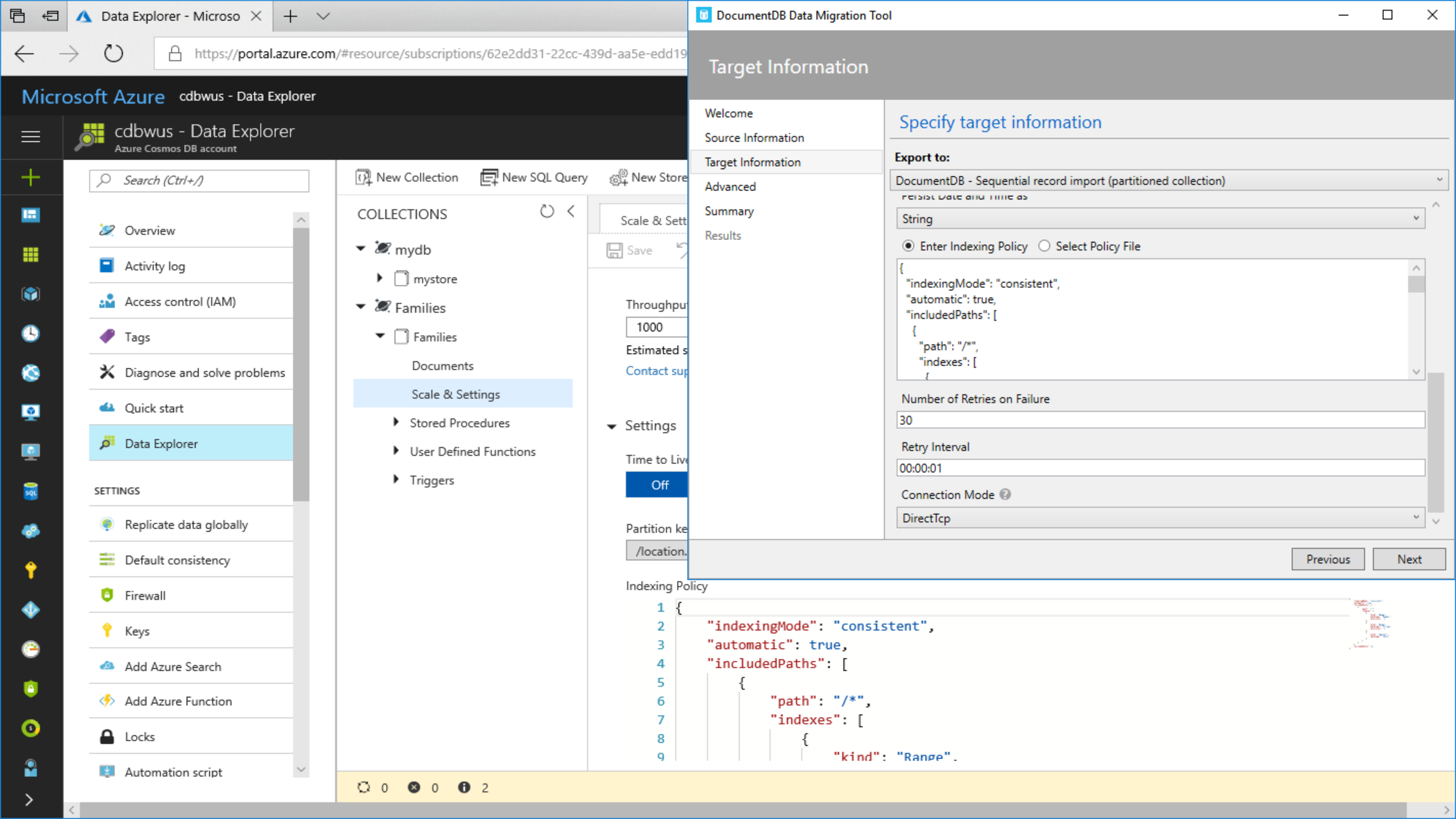
Points, polygons, line strings

Indexing Policies

Collection-wide policy

Established when creating a collection

Can be changed after collection is created



Export to: DocumentDB - Sequential record import (partitioned collection)

Persist Date and Time as: String

(●) Enter Indexing Policy ( ) Select Policy File

```
{
 "indexingMode": "consistent",
 "automatic": true,
 "includedPaths": [
  {
   "path": "/*",
   "indexes": [
    {
```

Number of Retries on Failure: 30

Retry Interval: 00:00:01

Connection Mode: DirectTcp

Indexing Policy

```
{
    "indexingMode": "consistent",
    "automatic": true,
    "includedPaths": [
        {
            "path": "/*",
            "indexes": [
                {
                    "kind": "Range",
```

Indexing Policies

Collection-wide policy

Established when creating a collection

Can be changed after collection is created

Automatic indexing

Can switch to manual

Can override on a per-document basis

Selective indexing

Include/exclude selected property paths

Individual properties (?)	Recursive properties (*)
`/name/?`	`/address/*`
`/address/state/?`	`/item/colors[]/*`

Indexing Policies

Collection-wide policy

Established when creating a collection

Can be changed after collection is created

Automatic indexing

Can switch to manual

Can override on a per-document basis

Selective indexing

Include/exclude selected property paths

Indexing modes

Consistent (synchronous)

Lazy (asynchronous)

Demo

Custom indexing

Users, Permissions, and Resource Tokens

Resource tokens vs. master key

Provides granular control over security

Create database users

Then create permissions for each user

Get resource token from permission

Read or **All** access to a single resource

(collection, document, stored procedures, triggers, user-defined functions)

Connect using resource tokens

Access will be granted based on all supplied resource tokens

Demo

Working with users and permissions

Summary

.NET SDK DocumentClient

- Databases
- Collections
- Documents
- Indexing policies
- Users and permissions