Module 03: Connect to Azure Cosmos DB SQL API with the SDK

Import and use the Azure Cosmos DB SQL API SDK

Understand the SDK

Class	Description
Microsoft.Azure.Cosmos.**CosmosClient**	Client-side logical representation of an Azure Cosmos DB account and the primary class used for the SDK
Microsoft.Azure.Cosmos.**Database**	Logically represents a database client-side and includes common operations for database management
Microsoft.Azure.Cosmos.**Container**	Logically represents a container client-side and includes common operations for container management

Import from package manager

```
dotnet add package Microsoft.Azure.Cosmos \
    --version 3.22.1
```

.NET project file

```
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <OutputType>Exe</OutputType>
    <TargetFramework>net6.0</TargetFramework>
  </PropertyGroup>
  <ItemGroup>
    <PackageReference Include="Microsoft.Azure.Cosmos" Version="3.22.1" />
  </ItemGroup>
</Project>
```

Connect to an online account - with connection string

In [ ]:

```
using Microsoft.Azure.Cosmos;

Console.WriteLine($"Using connectionString: {connectionString}");

CosmosClient client = new (connectionString);

client
```

Connect to an online account - with endpoint and key

Configure the Azure Cosmos DB SQL API SDK

In [ ]:

```
using Microsoft.Azure.Cosmos;

Console.WriteLine($"Using documentEndpoint: {documentEndpoint}");
Console.WriteLine($"Using primaryMasterKey: {primaryMasterKey}");

CosmosClient client = new (documentEndpoint, primaryMasterKey);
```

```
client
```

Read properties of the account

The **AccountProperties** class includes useful properties such as, but not limited to:

Property	Description
Id	Gets the unique name of the account
ReadableRegions	Gets a list of readable locations for the account
WritableRegions	Gets a list of writable locations for the account
Consistency	Gets the default consistency level for the account

In [ ]:

```
AccountProperties account = await client.ReadAccountAsync();

account
```

Interact with a database

In [ ]:

```
// Retrieve an existing database
Database database = client.GetDatabase("cosmicworks");

database
```

In [ ]:

```
// Create a new database
Database database = await client.CreateDatabaseAsync("cosmicworks");

// Note there can be an exception when the database already exists...
// Error: Microsoft.Azure.Cosmos.CosmosException : Response status code does not inc
```

In [ ]:

```
// Create database if it doesn't already exist
Database database = await client.CreateDatabaseIfNotExistsAsync("cosmicworks");

database
```

Interact with a container

In [ ]:

```
// Retrieve an existing container
Container container = database.GetContainer("products");

container
```

In [ ]:

```
// Create a new container
Container container = await database.CreateContainerAsync(
    "cosmicworks",
    "/categoryId",
    400
);

container
```

In [ ]:

```
// Create container if it doesn't already exist
Container container = await database.CreateContainerIfNotExistsAsync(
    "cosmicworks",
    "/categoryId",
    400
```

```
);

container
```

Implement client singleton

Each instance of the CosmosClient class has a few features that are already implemented on your behalf:

- Instances are already thread-safe
- Instances efficiently manage connections

Configure connectivity mode