

**Exploring Cosmos DB via the MongoDB API**

**Introduction**

During this lab, you will learn how to manage your data in Cosmos DB via the MongoDB API using the Azure Portal.

**Estimated Time**

25 minutes

**Objectives**

At the end of this lab, you will be able to:

         Use Azure Cosmos DB using the MongoDB API, create and query documents, and access existing collections

         Use MongoDB utilities to import and export data

Lab: Exploring Cosmos DB via the MongoDB API

During this lab, you will learn how to use Azure Cosmos DB using the MongoDB API, create and query documents, and access existing collections, and use MongoDB utilities to import and export data

**Exercise 1: Connect Using Mongo Shell to Create and Query Documents**

In this exercise, we will use Mongo Shell to query various documents using the MongoDB API in Azure Cosmos DB.

**Tasks**

**1.      Install Database Tooling**

1.       If you have already installed MongoDB Community Edition, you can skip this exercise.

2.       Download MongoDB Community Edition from this link:

<https://www.mongodb.com/download-center#community>

We will only use the MongoDB tools for this lab, but you can install the complete software suite.

3.       Once downloaded, install MongoDB Community Edition on your local machine.

4.       Once the installation has completed, navigate to the installation folder on your local machine.

For example, the Windows installation directory is typically

C:\%ProgramFiles%\MongoDB\Server\[version]\bin.

For an x64 version of Windows and 3.4 version of MongoDB, it would be:

C:\Program Files\MongoDB\Server\3.4\bin.

**2.      Create Account Resource**

1.       In a new window, sign in to the **Azure Portal** ([http://portal.azure.com](http://portal.azure.com/)).

2.       In the **Jumpbar (left navigation bar)**, click **Create a resource**, click **Databases**, and then click **Azure Cosmos DB**).

3.       In the **Azure Cosmos DB** blade, specify the desired configuration for the new Azure Cosmos DB account using the following steps:

                                                                           i.      In the **ID** box, enter a name to uniquely identify the account. When the **ID** is validated, a green check mark appears in the **ID** box.

                                                                         ii.      The **ID** value becomes the host name within the URI. The **ID** may contain only lowercase letters, numbers, and the '-' character, and must be between 3 and 50 characters.

                                                                       iii.      In the **API** section, select **MongoDB.**

                                                                       iv.      In the **Subscription** section, select the Azure subscription that you want to use for the account.

                                                                         v.      If your account has only one subscription, that subscription is selected by default.

                                                                       vi.      In the **Resource Group** section, select resource group **CosmosWorkshop**.

                                                                     vii.      Use **Location** to specify the geographic location closest to your current location in which to host your account.

4.       Once the new account options are configured, click the **Create** button to begin the deployment. To check the status of the deployment, check the **Notifications hub** at the top-right corner of your Azure portal.

5.       After the Azure Cosmos DB account is created, you will receive a notification in the **Notifications Hub** indicating that the deployment action is complete.

6.       Click the **Go to Resource** button in the notification to view the **Azure Cosmos DB** account resource.

**3.      Create Collection and Obtain Connection Information**

1.       In the Azure Cosmos DB account blade, click the **Data Explorer** option on the left-side of the blade.

2.       In the **Data Explorer** section, click the **New Collection** button at the top of the section.

3.       In the **Add Collection** dialog that appears, enter the following values:

                                                                           i.      **Database id**: contosodb

                                                                         ii.      **Collection id**: inventorycoll

                                                                       iii.      **Storage capacity**: Fixed (10 GB)

                                                                       iv.      **Throughput (400 - 10,000 RU/s)**: 400

4.       Click the **OK** button to create the new collection.

5.       Back in the Azure Cosmos DB account blade, click the **Connection String** option on the left-side of the blade.

6.       In the **Connection String** section, record the following values:

                                                                           i.      [Host]: Locate the **HOST** box and record the value.

                                                                         ii.      [Port]: Locate the **PORT** box and record the value.

                                                                       iii.      [Username]: Locate the **Username** box and record the value.

                                                                       iv.      [Password]: Locate the **Password** box and record the value.

**4.      Connect using Mongo Shell**

1.       Open a shell or command prompt at the MongoDB installation folder.

For example, the Windows installation directory is typically:

C:\%ProgramFiles%\MongoDB\Server\[version]\bin.

For an x64 version of Windows and 3.4 version of MongoDB, it would be:

C:\Program Files\MongoDB\Server\3.4\bin.

If you are using the latest update of Windows 10, the PowerShell IDE is the default shell/command prompt. It is recommended that you instead use the Command Prompt utility for these lab instructions.

2.       Enter the following command to open the **mongo** shell. For each of the authentication parameters, you should have recorded values earlier in this lab:

mongo --ssl --host [Host] --port [Port] --username [Username] --password [Password]

For example, if I have the following values:

[Host]: example.documents.azure.com

[Port]: 10250

[Username]: example

[Password]: L3u1xYOwE3RJBbrvCOgeAVGyk2N2AETdBdoV6w5ofoEoCh0gIvKauwYJkpc4rQIOiGPH8ZGxdaOMJUtSOoOMOL==

Then my command will look like this:

mongo --ssl --host example.documents.azure.com --port 10250 --username example --password L3u1xYOwE3RJBbrvCOgeAVGyk2N2AETdBdoV6w5ofoEoCh0gIvKauwYJkpc4rQIOiGPH8ZGxdaOMJUtSOoOMOL==

**5.      Access Collection**

1.       In the **mongo** shell, run the following command to list all databases in your account:

   show dbs

2.       Run the following command to switch to the **contosodb** database:

   use contosodb

3.       Run the following command to view all documents in the **inventorycoll** collection within the current database:

                  db.inventorycoll.find()

You should not see any documents since we have not created any yet.

**6.      Create Documents**

1.       Run the following command to insert a single document into the collection with two properties:

db.inventorycoll.insert({ "product": "wheel", "type": "part" })

2.       Run the following command to insert a single complex document into the collection that includes an embedded document:

db.inventorycoll.insert({ "product": "light", "type": "safety", "features": { "lumens": 600, "battery": "AAA" } })

3.       Run the following command to insert multiple simple documents into the collection:

db.inventorycoll.insertMany([{ "product": "warranty" }, { "product": "servicePlan" }])

4.       Run the following command to count the number of documents in the collection:

db.inventorycoll.count()

**7.      Query Documents**

1.       Run the following command to find all documents in the collection:

db.inventorycoll.find()

2.       Run the following command to find only documents with a type property set to a value of part:

   db.inventorycoll.find({ "type": "part" })

3.       Run the following command to find only documents where the type property exists:

   db.inventorycoll.find({ type: { $exists: false } })

4.       Run the following command to find only documents with an embedded document set as the value of the features property and a child property named lumens with a value greater than 500:

   db.inventorycoll.find({ "features.lumens": { $gt: 500}})

5.       Close the command prompt window running the mongo shell utility.

*Exercise 1 has been completed.*

**Exercise 2: Import JSON Formatted Data and Export to CSV Format**

This exercise shows how to import JSON formatted data and export to CSV format using Mongo utilities.

**Tasks**

**1.      Download Example Data File and Import JSON Formatted Data**

1.       Download the Compressed folder containing a single JSON file from this link:  
[input\_docs.json](https://prod-edxapp.edx-cdn.org/assets/courseware/v1/e6330728d6981691f2c78a3f4cad7cf0/asset-v1:Microsoft+DAT237x+1T2018+type@asset+block/04-docs.json)

2.       On your local machine, save the downloaded [File] to a directory that you will be able to access later in this lab. Record the name of this location so you can reference it in future steps in this lab.

3.      Open a shell or command prompt at the MongoDB installation folder.

If you are using the latest update of Windows 10, the PowerShell IDE is the default shell/command prompt. It is recommended that you instead use the Command Prompt utility for these lab instructions.

For example, the Windows installation directory is typically:

C:\%ProgramFiles%\MongoDB\Server\[version]\bin.

For an x64 version of Windows and 3.4 version of MongoDB, it would be:

C:\Program Files\MongoDB\Server\3.4\bin.

4.      Enter the following command to run the **mongoimport** utility. For each of the authentication parameters, you should have recorded values earlier in this lab:

mongoimport --ssl --host [Host] --port [Port] --username [Username] --password [Password] --db contosodb --collection inventorycoll --file [File]

For example, if I have the following values:

[Host]: example.documents.azure.com

[Port]: 10250

[Username]: example

[Password]: L3u1xYOwE3RJBbrvCOgeAVGyk2N2AETdBdoV6w5ofoEoCh0gIvKauwYJkpc4rQIOiGPH8ZGxdaOMJUtSOoOMOL==

[File]: C:\Users\ExampleUser\Download\input\_docs.json

Then my command will look like this:

mongoimport --ssl --host example.documents.azure.com --port 10250 --username example --password L3u1xYOwE3RJBbrvCOgeAVGyk2N2AETdBdoV6w5ofoEoCh0gIvKauwYJkpc4rQIOiGPH8ZGxdaOMJUtSOoOMOL== --db contosodb --collection inventorycoll --file C:\Users\ExampleUser\Download\input\_docs.json

**2.      View Imported Data**

1.      Return to the browser tab/windows showing the Azure portal.

2.      In the Azure Cosmos DB account blade, click the **Data Explorer** option on the left-side of the blade.

3.      In the **Data Explorer** section, click the **New Mongo Shell** button at the top of the section.

4.      In the *shell* tab, run the following command to view your imported documents:

db.inventorycoll.find()

**3.      Export to CSV Format**

1.      Open a shell or command prompt at the MongoDB installation folder on your local machine.

If you are using the latest update of Windows 10, the PowerShell IDE is the default shell/command prompt. It is recommended that you instead use the Command Prompt utility for these lab instructions.

2.      Enter the following command to run the **mongoexport** utility. For each of the authentication parameters, you should have recorded values earlier in this lab:

mongoexport --ssl --host [Host] --port [Port] --username [Username] --password [Password] --db contosodb --collection inventorycoll --out [File] --type csv --fields product,type

For example, if I have the following values:

[Host]: example.documents.azure.com

[Port]: 10250

[Username]: example

[Password]: L3u1xYOwE3RJBbrvCOgeAVGyk2N2AETdBdoV6w5ofoEoCh0gIvKauwYJkpc4rQIOiGPH8ZGxdaOMJUtSOoOMOL==

[File]: C:\Users\ExampleUser\Download\output\_docs.csv

Then my command will look like this:

mongoimport --ssl --host example.documents.azure.com --port 10250 --username example --password L3u1xYOwE3RJBbrvCOgeAVGyk2N2AETdBdoV6w5ofoEoCh0gIvKauwYJkpc4rQIOiGPH8ZGxdaOMJUtSOoOMOL== --db contosodb --collection inventorycoll --out C:\Users\ExampleUser\Download\output\_docs.csv --type csv --fields product,type

4.      Open the generated file on your local machine to view the export results.

*Exercise 2 has been completed.*