

**Exploring Cosmos DB via the Table API**

**Introduction**

During this lab, you will learn how to create and query entities in Azure Cosmos DB Table API.

**Estimated Time**

15 minutes

**Objectives**

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

         Use the Table API in Cosmos DB to create tables and entities, and be able to query them.

Lab: Exploring Cosmos DB via the Table API

During this lab, you will learn how to create and query entities in Azure Cosmos DB Table API.

**Exercise 1: Create and Query Entities**

In this hands-on exercise, we will use Azure Cosmos Db to create entities in the Table API.

**Tasks**

**1.      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.

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.

                                                                         ii.      In the **API** section, select **Azure Table.**

                                                                       iii.      In the **Subscription** section, select the Azure subscription that you want to use for the account. If your account has only one subscription, that subscription is selected by default.

                                                                       iv.      In the **Resource Group** section, choose resource group for your account **CosmosWorkshop.**

                                                                         v.      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. If you cannot find the notification, click the **bell** icon in the Azure portal to view your list of notifications.

**2.      Create Entities**

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 Table**. Chose a **Table ID** (for example: Collection 1)with a storage capacity of **10 GB** and throughput of **400 RU/s**. Click **OK**.

3.       In the *entities* tab, click the **Add Entity** button. Add the following values and click **Add Entity** to confirm. You may need to click the **Add Property** button to add additional properties.

|  |  |  |
| --- | --- | --- |
| **Property Name** | **Type** | **Value** |
| PartitionKey | String | students |
| RowKey | String | bstonge |
| Age | Int32 | 8 |

4.       Repeat step 3 above with the following values.

|  |  |  |
| --- | --- | --- |
| **Property Name** | **Type** | **Value** |
| PartitionKey | String | Students |
| RowKey | String | Vgray |
| FirstName | String | Victoria |
| LastName | String | Gray |
| Age | Int32 | 9 |

5.       Repeat step 3 above with the following values.

|  |  |  |
| --- | --- | --- |
| **Property Name** | **Type** | **Value** |
| PartitionKey | String | teachers |
| RowKey | String | sgarner |
| LastName | String | Garner |
| IsAdministrator | Boolean | True |

6.      View the three new entities in your table within the **entities** tab.

**3.      Query Entities**

1.       Back in the *entities* tab, click the **Query Text** button to show the OData filter text. The current filter is an empty string, so all table entities are returned.

2.       Click the **Query Builder** button to show the query editor.

3.       Click the **Add new clause** button to create a query clause.

4.       In the new query clause, fill out the horizontal form with the following values:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **And/Or** | **Field** | **Type** | **Operator** | **Value** |
|  | PartitionKey | String | = | students |

5.       Click the **Query Text** button to view the OData filter text generated by the *Query Builder* tool.

6.       Click the **Run** button to execute the query. This query returns records within the **students** partition.

7.       View the updated table with the results of the query.

8.       Click the **Query Builder** button to show the query editor.

9.       Click the **Add new clause** button to create a query clause.

10.   In the new query clause, fill out the horizontal form with the following values:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **And/Or** | **Field** | **Type** | **Operator** | **Value** |
| And | RowKey | String | = | vgray |

11.   Click the **Query Text** button to view the OData filter text generated by the *Query Builder* tool.

12.   Click the **Run** button to execute the query. This query uses the **row** and **partition keys** to return a unique record.

13.   Click the **Query Builder** button to show the query editor.

14.   Click the **X** button to delete the existing query clause that filters data using the value of the **RowKey** field.

15.   Click the **Add new clause** button to create a query clause.

16.   In the new query clause, fill out the horizontal form with the following values:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **And/Or** | **Field** | **Type** | **Operator** | **Value** |
|  | Age | Int32 | < | 9 |

17.  Click the **Query Text** button to view the OData filter text generated by the *Query Builder* tool.

18.  Click the **Run** button to execute the query. This query returns all entities within the **students** partition that meets specific criteria.

19.  Click the **Query Builder** button to show the query editor.

20.  Click the **X** button to delete the existing query clause that filters data using the value of the **PartitionKey** field.

21.  Click the **Query Text** button to view the OData filter text generated by the *Query Builder* tool.

22.  Click the **Run** button to execute the query. This is a cross-partition query. This is NOT an efficient query because the query must traverse data on all your table partitions.

*Exercise 1 has been completed.*