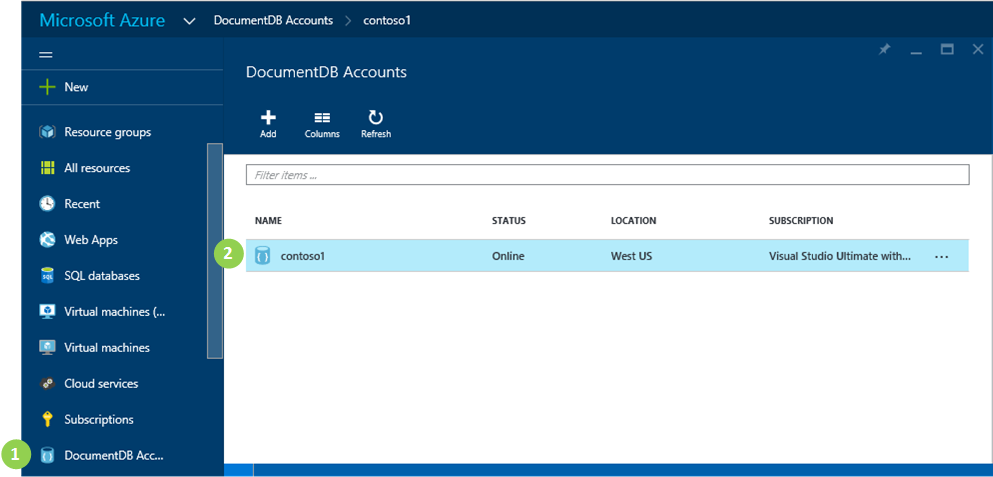
# How to create a DocumentDB collection using the Azure portal

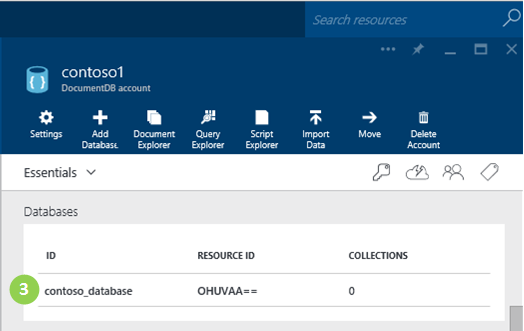
To use Microsoft Azure DocumentDB, you must have a [DocumentDB account](/documentation/articles/documentdb-create-account), a [database](/documentation/articles/documentdb-create-database), a collection, and documents. This topic describes how to create a DocumentDB collection in the Azure portal.

Not sure what a collection is? See [What is a DocumentDB collection?](#what-is-a-documentdb-collection)

1. In the [Azure portal](https://portal.azure.com/), in the Jumpbar, click **DocumentDB Accounts**. If **DocumentDB Accounts** is not visible, click **Browse** and then click **DocumentDB Accounts**.

* 
* Screen shot highlighting the DocumentDB Accounts in the Jumpbar, the account in the DocumentDB Accounts blade, and the database in the DocumentDB account blade, in the Databases lens

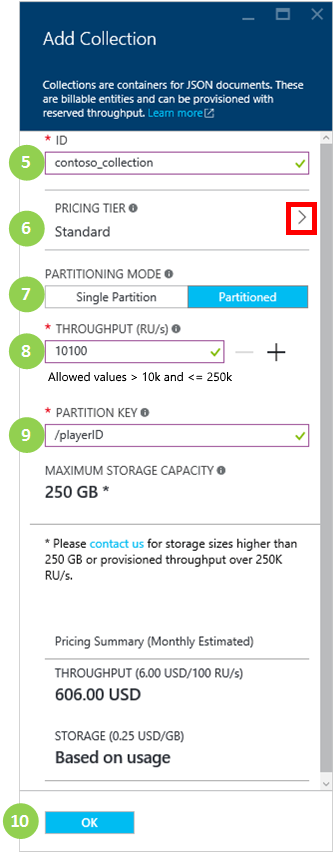
1. In the **DocumentDB Accounts** blade, select the account in which to add a collection. If you don’t have any accounts listed, you’ll need to [create a DocumentDB account](/documentation/articles/documentdb-create-account).
2. In the **DocumentDB account** blade for the selected account, scroll down to the **Databases** lens, and then select the database in which to add a collection.

* 
* Screen shot highlighting the DocumentDB Accounts in the Jumpbar, the account in the DocumentDB Accounts blade, and the database in the DocumentDB account blade, in the Databases lens

1. In the **Database** blade, click **Add collections**.

* 
* Screen shot highlighting the Add Collection button on the Database blade, the settings on the Add Collection blade, and the OK button - Azure portal for DocumentDB - Cloud based database creator for NoSQL JSON databases

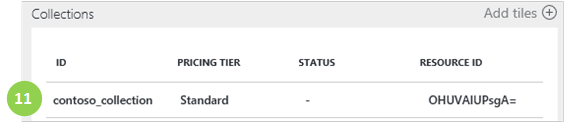
1. In the **Add Collection** blade, in the **ID** box, enter the ID for your new collection. Collection names must be between 1 and 255 characters, and cannot contain / \ # ? or a trailing space. When the name is validated, a green check mark appears in the ID box.

* 
* Screen shot highlighting the Add Collection button on the Database blade, the settings on the Add Collection blade, and the OK button - Azure portal for DocumentDB - Cloud based database creator for NoSQL JSON databases

1. If you are going to create a collection with a single partition, click the arrow to select a pricing tier, then click View All, then select a performance level for the collection. For more information about the performance levels available, see [Performance levels in DocumentDB](/documentation/articles/documentdb-performance-levels). Each collection you create is a billable entity.

* If you are going to create a partitioned collection, skip to step 7 as you do not need to select a pricing tier.

1. Select a **Partitioning mode** for the collection, either **Single Partition** or **Partitioned**. A single partition has a reserved storage capacity of 10GB, and can have throughput levels from 400-10,000 request units/second. A partitioned collection can scale to handle 250GB of storage over multiple partitions, and can have throughput levels from 10,100-250,000 request units/second.
2. Select the **Throughput** for the partitioned collection. One Request Unit (RU) corresponds to the throughput of a read of a 1KB document. For more information about request units, see [Request units](/documentation/articles/documentdb-request-units).
3. If you are creating a partitioned collection, select the **Partition Key** for the collection. Selecting the correct partition key is important in creating a performant collection. For more information on selecting a partition key, see [Partitioning data in DocumentDB](/documentation/articles/documentdb-partition-data).
4. Click **OK** at the bottom of the screen to create the new collection.
5. The new collection now appears in the **Collections** lens on the **Database** blade.

* 
* Screen shot of the new collection in the Database blade - Azure portal for DocumentDB - Cloud based database creator for NoSQL JSON databases

## What is a DocumentDB collection?

A collection is a container of JSON documents and the associated JavaScript application logic. A collection is a billable entity, where the [cost](/documentation/articles/documentdb-performance-levels) is determined by the provisioned throughput of the collection. Collections can span one or more partitions/servers and can scale to handle practically unlimited volumes of storage or throughput.

Collections are automatically partitioned into one or more physical servers by DocumentDB. When you create a collection, you can specify the provisioned throughput in terms of request units per second and a partition key property. The value of this property will be used by DocumentDB to distribute documents among partitions and route requests like queries. The partition key value also acts as the transaction boundary for stored procedures and triggers. Each collection has a reserved amount of throughput specific to that collection, which is not shared with other collections in the same account. Therefore, you can scale out your application both in terms of storage and throughput.

Collections are not the same as tables in relational databases. Collections do not enforce schema, in fact DocumentDB does not enforce any schemas, it’s a schema-free database. Therefore you can store different types of documents with diverse schemas in the same collection. You can choose to use collections to store objects of a single type like you would with tables. The best model depends only on how the data appears together in queries and transactions.

## Other ways to create a DocumentDB collection

Collections do not have to be created using the portal, you can also create them using the [DocumentDB SDKs](documentdb-sdk-dotnet.md) and the REST API.

* For a C# code sample, see the [C# collection samples](/documentation/articles/documentdb-dotnet-samples#collection-examples).
* For a Node.js code sample, see the [Node.js collection samples](/documentation/articles/documentdb-nodejs-samples#collection-examples).
* For a Python code sample, see [Python collection samples](/documentation/articles/documentdb-python-samples#collection-examples).
* For a REST API sample, see [Create a Collection](https://msdn.microsoft.com/library/azure/mt489078.aspx).

## Troubleshooting

If **Add Collection** is disabled in the Azure portal, that means your account is currently disabled, which normally occurs when all the benefits credits for the month are used.

## Next steps

Now that you have a collection, the next step is to add documents or import documents into the collection. When it comes to adding documents to a collection, you have a few choices:

* You can [add documents](/documentation/articles/documentdb-view-json-document-explorer) by using the Document Explorer in the portal.
* You can [import documents and data](/documentation/articles/documentdb-import-data) by using the DocumentDB Data Migration Tool, which enables you to import JSON and CSV files, as well as data from SQL Server, MongoDB, Azure Table storage, and other DocumentDB collections.
* Or you can add documents by using one of the [DocumentDB SDKs](/documentation/articles/documentdb-sdk-dotnet). DocumentDB has .NET, Java, Python, Node.js, and JavaScript API SDKs. For C# code samples showing how to work with documents by using the DocumentDB .NET SDK, see the [C# document samples](/documentation/articles/documentdb-dotnet-samples#document-examples). For Node.js code samples showing how to work with documents by using the DocumentDB Node.js SDK, see the [Node.js document samples](/documentation/articles/documentdb-nodejs-samples#document-examples).

After you have documents in a collection, you can use [DocumentDB SQL](/documentation/articles/documentdb-sql-query) to [execute queries](/documentation/articles/documentdb-sql-query#executing-queries) against your documents by using the [Query Explorer](/documentation/articles/documentdb-query-collections-query-explorer) in the portal, the [REST API](https://msdn.microsoft.com/library/azure/dn781481.aspx), or one of the [SDKs](/documentation/articles/documentdb-sdk-dotnet).