



# Azure Cosmos DB Adapter for SAP Integration Suite

Version 1.0.1 – February 2025

# Contents

1	Introduction .....	3
1.1	Objective .....	3
1.2	Coding Samples .....	3
1.3	Internet Hyperlinks .....	3
1.4	Overview .....	3
1.5	Features .....	4
2	Installation and Configuration.....	5
2.1	Adapter Installation on Cloud Foundry .....	5
2.1.1	Prerequisites .....	5
2.1.2	Procedure .....	5
2.1.3	Adapter Installation by creating a New Integration Flow .....	5
2.1.4	Adapter Installation without Creating a New Integration Flow.....	6
2.2	Monitor the Deployment Status .....	7
3	Getting Started: Azure Cosmos DB Adapter.....	8
3.1	Architecture Overview .....	8
3.2	Application Configuration .....	8
3.3	Authentication.....	9
3.3.1	Creating Credentials in Security Material .....	9
3.4	Supported version .....	10
4	Azure Cosmos DB Adapter Configuration.....	11
4.1	General.....	11
4.2	Connection.....	12
4.3	Processing .....	13
5	Azure Cosmos DB Supported Operations .....	15
5.1	Single Processing Mode.....	15
5.1.1	Create.....	15
5.1.2	Delete .....	16
5.1.3	Get.....	17
5.1.4	List .....	17

5.1.5	Query .....	18
5.1.6	Update .....	19
5.1.7	Replace .....	20
5.1.8	Get Partition Key Ranges .....	21
5.2	Bulk Processing Mode .....	22
5.2.1	Create .....	22
5.2.2	Replace .....	22
5.2.3	Upsert .....	23
5.2.4	Delete .....	24
5.2.5	Patch .....	25

# 1 Introduction

## 1.1 Objective

This is the official guide for the Azure Cosmos DB Adapter for SAP Integration Suite. This guide covers all relevant information for integration developers to start working with the Azure Cosmos DB adapter. Read this guide carefully before using the Adapter.

## 1.2 Coding Samples

Any software coding and/or code lines/strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended to better explain and visualize the syntax and phrasing rules of certain coding. The correctness and completeness of the Code given herein are not guaranteed.

## 1.3 Internet Hyperlinks

The documentation may contain hyperlinks to the Internet. These hyperlinks are intended to serve as a hint about where to find related information. The availability and the correctness of this related information or the ability of this information to serve a particular purpose are not warranted.

## 1.4 Overview

Azure Cosmos database is a platform as a service and a cloud-based NoSQL database by Microsoft Azure. It is a fully managed, globally distributed JSON-based database that supports high-level horizontal scaling, replication, and partitioning.


The Azure Cosmos DB lets you use the API of your choice to connect and store data. In Azure Cosmos DB you can provision throughput on the Database and the Container/Collection level too. The Azure Cosmos DB adapter can help you exchange the data between Azure Cosmos DB and other systems.

## 1.5 Features

- Provides the ability to interact with entities such as Database, Container, and Document.
- Supports standard CRUD (Create, Read, Update, and Delete) operations in Azure Cosmos DB.
- Supports operations such as list, query, and replace on various entities in Azure Cosmos DB.
- Efficiently handles large sets of documents with bulk operation capabilities.
- Supports multiple API versions of Azure Cosmos DB.
- Offers a range of authentication control options from full access to resource-based access using Master Key Token and Resource Token.
- Uses dynamic values and parameters to customize integration flows seamlessly with SAP Integration Suite headers and properties.

# 2 Installation and Configuration

This section details the file(s) available as part of the installation package and the prerequisites to configure the Azure Cosmos DB adapter.

 The Azure Cosmos DB adapter is available as part of your SAP Integration Suite license.

## 2.1 Adapter Installation on Cloud Foundry


Before the adapter can be used in the Cloud Foundry environment, it must be deployed to the SAP Integration Suite tenant.

### 2.1.1 Prerequisites

To deploy the Azure Cosmos DB adapter, you must have access to the *SAP Integration Suite* license.

### 2.1.2 Procedure

You can deploy the adapter using the following methods:

 The following installation procedure is compatible with Apache Camel 2, Apache Camel 3, and Edge Integration Cell (EIC) platform.

### 2.1.3 Adapter Installation by creating a New Integration Flow

The Azure Cosmos DB adapter is available for selection in the receiver adapter list and can be deployed in the **Design** tab directly as you use it in an Integration flow.



#### **Purpose**

To install an adapter for use in your Integration flow.


#### **Procedure**

Go to **Design** workspace and select the integration package where you want to create a new Integration flow.

1. Click **Edit** to make the package editable.
2. Go to the **Artifacts** tab. Click **Add** and select **Integration Flow**.

3. Enter **Name** and **ID** for your flow. Additionally, select **Runtime Profile** from the drop-down and choose **Sender** and **Receiver** systems from the list . Finally, click **Add** to create the integration flow.
4. Go to the newly created integration flow and click **Edit** to make it editable.
5. In the integration flow, click **End** to add a **Connector**  between the **End** and the **Receiver Box**.  
A drop-down with the available adapters appears. The **Azure Cosmos DB** adapter should show up in the list.
6. Select the **Azure Cosmos DB** adapter from the list. The adapter is now imported which *triggers* an adapter deployment. Once Azure Cosmos DB Adapter is deployed, a success message is displayed.  
After the above steps are done, the Azure Cosmos DB Adapter is successfully deployed in your Design workspace of the SAP Integration Suite tenant.

## 2.1.4 Adapter Installation without Creating a New Integration Flow

 The following procedure explains how the Azure Cosmos DB adapter is migrated from the Discover workspace to the Design workspace of the SAP Integration tenant.

This method is useful for scenarios where integration flow packages are migrated from development to a higher environment such as Production.

The Azure Cosmos DB adapter can be imported into the Design workspace without creating an integration flow. Use the Transport Management Service (TMS) to import/transport the Azure Cosmos DB adapter to a higher environment. Alternatively, If the TMS is not available in the landscape, the adapter package can be imported to the Design workspace by copying it from the Discover workspace.

### Purpose

To copy the integration package from the Discover workspace and import the Azure Cosmos DB adapter to the Design workspace, follow these steps:

### Procedure

1. Go to **Discover** workspace.
2. In the search box, search for **AzureCosmosDB adapter for SAP Integration Suite** package.
3. Select the package and click **Copy**. This copies the package from the Discover workspace to the Design workspace.
4. Go to Design workspace and select the copied **AzureCosmosDB adapter for SAP Integration Suite** package.

5. In the **Actions** tab of the selected package, click **Deploy**. This completes the adapter deployment to the Design workspace.

## 2.2 Monitor the Deployment Status

After the adapter deployment is complete, you can check the status in the **Monitor** section.

### Purpose

To check the status of the deployed adapter.

### Procedure

1. Under the **Monitor** tab, click **Integrations and APIs**. This opens the **Overview** page.
2. On the **Overview** page, go to **Manage Integration Content** section and click **All**. This opens **Integration Content** page with a list of all the deployed adapters.
3. Here, you can check and confirm the deployment status of your adapter.

The screenshot displays the 'Manage Integration Content' interface. On the left, a search bar contains 'AzureCosmosDB'. Below it, a table lists the integration content:

Name	Status
AzureCosmosDB	Started
Integration Adapter	

On the right, the details for 'AzureCosmosDB' are shown, including 'Deployed On: Nov 21, 2024, 14:09:32', 'ID:', 'Deployed By:', 'Version: 1.0.0', and 'Package:'. A green status message at the bottom right states: 'The Integration Adapter is deployed successfully.'

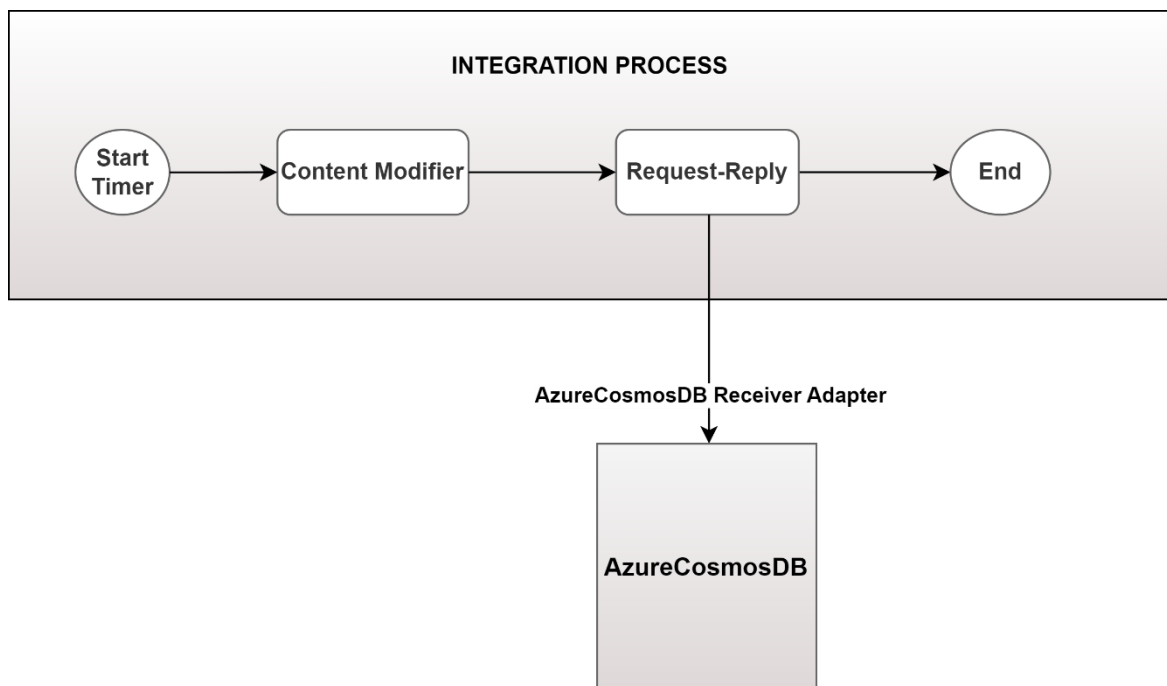


# 3 Getting Started: Azure Cosmos DB Adapter

This section explains how to configure the Azure Cosmos DB adapter for SAP Integration Suite. You can find information about adapter architecture, application configuration, authentication, and supported versions for Azure Cosmos DB Adapter.

## 3.1 Architecture Overview

The Azure Cosmos DB adapter is designed to function as a receiver adapter. In such a scenario where the Azure Cosmos DB Adapter is used as a receiver adapter, SAP Integration Suite acts as the initiator of the calls.



## 3.2 Application Configuration

You can connect to Azure Cosmos DB via the Azure Cosmos DB Adapter using your user credentials. Before you begin, the following links can be helpful to set up your Azure Cosmos DB application.

To create an Azure Cosmos DB Account, see [Create an Azure Cosmos DB account](#).

## 3.3 Authentication

This section details the authentication mechanism supported by the Azure Cosmos DB Adapter in SAP Integration Suite.

You can find a Master Key in the 'Keys' section of the account or you can have a Resource Token for a user. You must use the Secure Parameter artifact to safely store the Access Key and Secret Key. These security artifacts can then be accessed in the adapter using aliases.

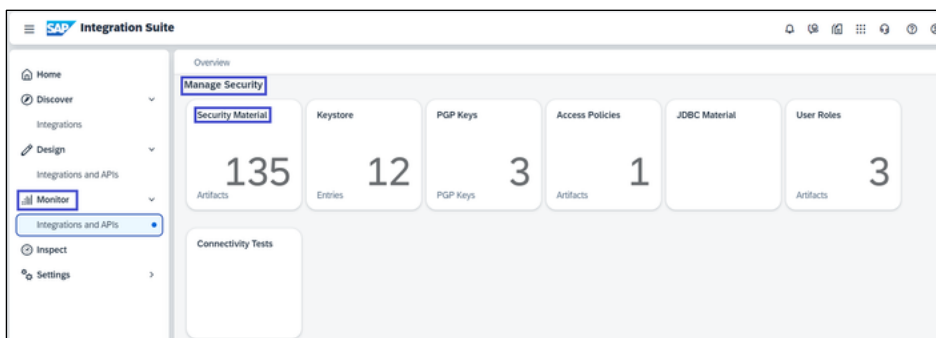
### 3.3.1 Creating Credentials in Security Material

#### Purpose

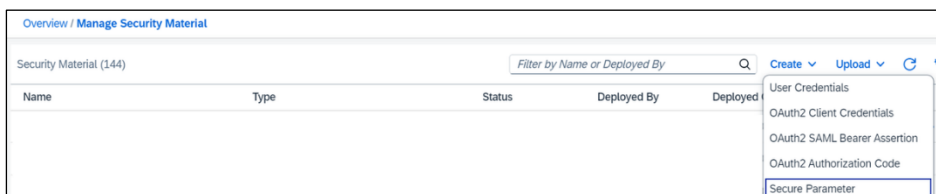
To create credentials in Security Material for Secure Parameter.

#### Procedure

1. In SAP Integration Suite, navigate to **Monitor > Integrations and APIs**. This opens the **Overview** page.
2. On the **Overview** page, go to **Manage Security** section and click **Security Material**.



3. On **Manage Security Material** page, click **Create** to select **Secure Parameter** from the dropdown.



4. In the **Create Secure Parameter** popup, provide the below details.

Parameter	Description
<b>Name</b>	Specify the name of the security artifact. The artifact name is used as an alias for the confidential data.
<b>Description</b>	Enter a description for the artifact (optional).
<b>Secure Parameter</b>	Enter the confidential value of the attribute. The permissible length of the secure parameter for Cloud Foundry is a maximum of 4096 characters.
<b>Repeat Secure Parameter</b>	Repeat the confidential value of the attribute.

5. Click **Deploy** to complete the process.

When you refresh the **Manage Security Material** page, the new artifact is displayed (with Type **Secure Parameter**) in the artifact table.

## 3.4 Supported version

The supported API versions are listed in the table below.

- 2018-12-31
- 2018-09-17
- 2018-08-31
- 2018-06-18
- 2017-11-15
- 2017-05-03
- 2017-02-22
- 2017-01-19
- 2016-07-11
- 2015-12-16
- 2015-08-06
- 2015-06-03
- 2015-04-08
- 2014-08-21



The version field is editable and can be updated to use the latest API version available in Azure Cosmos DB.

# 4 Azure Cosmos DB Adapter Configuration

This section describes the parameters to be configured for your Azure Cosmos DB adapter. You need to configure the **General**, **Connection**, and **Processing** tabs. A description and example usage for every field has been added.

## Receiver Adapter

In this section, you will learn how to configure the Azure Cosmos DB receiver adapter. On selecting the Azure Cosmos DB adapter from the list of adapters, you must configure the **General**, **Connection**, and **Processing** tabs.

### 4.1 General

The General tab provides an overview of basic adapter information including **Channel** and **Adapter** details.

The screenshot shows the configuration window for the AzureCosmosDB adapter. The window title is 'AzureCosmosDB'. It has three tabs: 'General' (selected), 'Connection', and 'Processing'. Below the tabs, there is a 'Name' field with the value 'AzureCosmosDB'. The interface is divided into two sections: 'CHANNEL DETAILS' and 'ADAPTER DETAILS'. Under 'CHANNEL DETAILS', there are three fields: 'Direction' (Receiver), 'System' (Receiver), and 'Description' (empty). Under 'ADAPTER DETAILS', there are three fields: 'Adapter Type' (AzureCosmosDB), 'Transport Protocol' (HTTPS), and 'Message Protocol' (REST NO-SQL).

Only the Name and Description fields are editable.

Parameter	Description
<b>Name</b>	Name of the adapter integration flow
<b>Description</b>	Description of the adapter

## 4.2 Connection



The Connection tab contains connection and authentication parameters for Azure Cosmos DB.


The Security artifact created in the previous section ([Creating Credentials in Security Material](#)) should be used in the **Connection tab** of the Adapter as shown in the figure below.

The screenshot shows the 'AzureCosmosDB' configuration window with the 'Connection' tab selected. The 'CONNECTION DETAILS' section includes the following fields:

- Address:** A text input field with a red asterisk, containing a blurred address.
- Authentication:** A dropdown menu set to 'Master Key Token'.
- Key Alias:** A text input field with a red asterisk, containing a blurred alias.
- Reuse Connection:** An unchecked checkbox.
- Connection Timeout (in ms):** A text input field containing '60000'.
- Response Timeout (in ms):** A text input field containing '60000'.

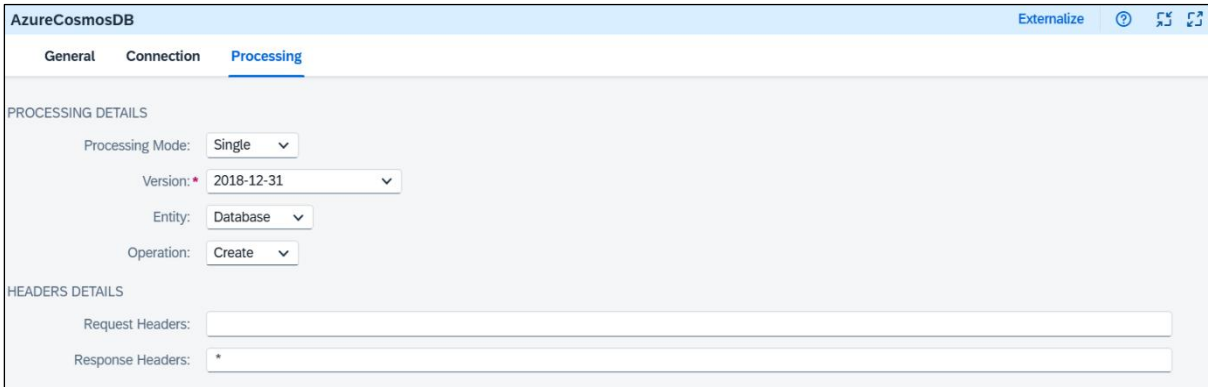
The connection tab contains the following fields:

Parameter	Description
<b>Address</b>	Specify the address with port of the host. Example: <code>https://{databaseaccount}.documents.azure.com:443/</code>
<b>Authentication</b>	Select the type of key for connecting to the Azure Cosmos DB account. <ul style="list-style-type: none"> <li>• Master Key Token</li> <li>• Resource Token</li> </ul>
<b>Key Alias</b>	Specify the Master key alias for connecting to the Azure Cosmos DB account.
<b>Token Alias</b>	Specify the Resource token alias for connecting to the Azure Cosmos DB account.
<b>Reuse Connection</b>	Enable this property to reuse the connection.  This field is applicable to a scenario when <b>Processing Mode</b> is set to <b>Single</b> .
<b>Connection Timeout (in ms)</b>	Specify the connection timeout in milliseconds. The timeout controls the maximum waiting time for the connection to Cosmos DB.  This field is applicable to a scenario when <b>Processing Mode</b> is set to <b>Single</b> .

Parameter	Description
<b>Response Timeout (in ms)</b>	<p>Specify the response timeout in milliseconds. The timeout controls the maximum waiting time until a response message is received.</p> <p> This field is applicable to a scenario when <b>Processing Mode</b> is set to <b>Single</b>.</p>


## 4.3 Processing






The Processing tab lists all the operations that can be performed on the database through the adapter.



The screenshot shows the 'Processing' tab in the Azure Cosmos DB configuration interface. It includes the following details:

- Processing Mode:** Single
- Version:** 2018-12-31
- Entity:** Database
- Operation:** Create
- Request Headers:** (empty text box)
- Response Headers:** \*

Parameter	Description
<b>Processing Details</b>	
<b>Processing Mode</b>	<p>Select the type of processing mode.</p> <ul style="list-style-type: none"> <li>• Single</li> <li>• Bulk</li> </ul>
<b>Version</b>	<p>Select the API version.</p> <p> This field is editable and can be updated to use the latest API version available in Azure Cosmos DB.</p>
<b>Entity</b>	<p>Select the entity for the operation to be performed.</p> <ul style="list-style-type: none"> <li>• Database</li> <li>• Container</li> <li>• Document</li> </ul>
<b>Operation</b>	<p>Select the type of operation to be performed.</p>

Parameter	Description
<b>Patch Operation</b>	Select the type of bulk patch operation for Documents.
<b>Database</b>	Specify the ID of the Database.  This value can be a valid string.
<b>Container</b>	Specify the ID of the Container.  This value can be a valid string.
<b>Document</b>	Specify the ID of the Document.  This value can be a valid string.
<b>Partition Key</b>	Specify the partition key for the operation.  <ul style="list-style-type: none"> <li>This field is mandatory only for some specific versions. For more information, see <a href="#">Azure Cosmos DB documentation</a>.</li> <li>This is required while performing operation in Single processing mode.</li> </ul>
<b>Partition Key Path</b>	Specify the path of the partition key. Example: /partitionkey
<b>Path</b>	Specify the path of the bulk patch operation. Example: /address/city
<b>Header Details</b>	
<b>Request Headers</b>	Enter a list of custom headers, separated by a pipe ( ), to send to the target system. By default, no custom headers are sent. Use an asterisk(*) to send all custom headers to the target system. Alternatively, you can dynamically pass on the values by defining a property that includes a list of headers.  This field is applicable to a scenario when <b>Processing Mode</b> is set to <b>Single</b> .
<b>Response Headers</b>	Enter a list of headers coming from the target system's response, separated by a pipe ( ), to be received in the message. Use an asterisk(*) to receive all the headers from the target system, which is also the default value.

# 5 Azure Cosmos DB Supported Operations

This section lists and describes some of the operations supported by the Azure Cosmos DB adapter. There are two processing modes available to execute an operation which are **Single** and **Bulk** by selecting the option in the dropdown.

## 5.1 Single Processing Mode

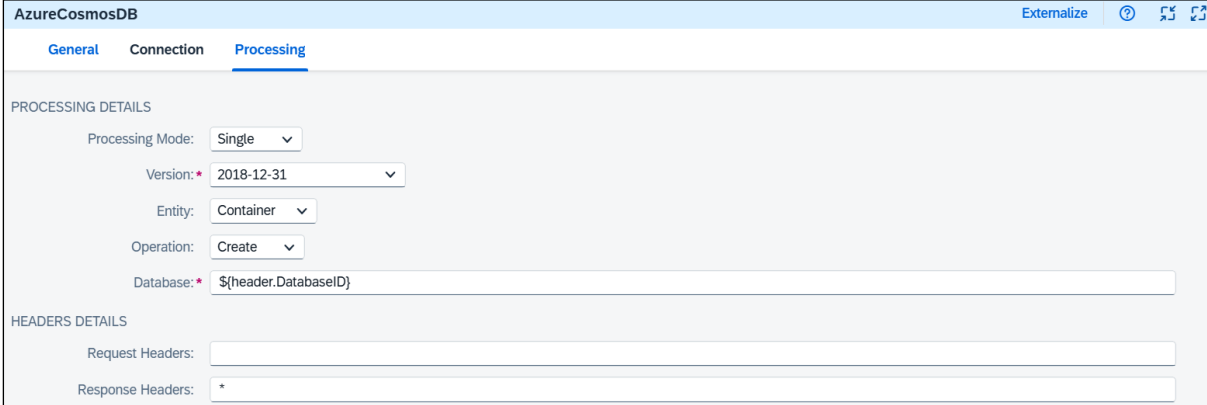
The following operations for single processing pertain to three entities within Azure Cosmos DB: **Database**, **Container**, and **Document**.

### 5.1.1 Create

The **Create** operation can be used to create a Database, Container, or Document in Azure Cosmos DB.

You can set the Processing mode to **Single**, select the API **Version**, choose an **Entity** type from options such as **Database**, **Container**, and **Document**, and select operation as **Create**. For more information about the configuration fields, see [Processing Tab](#).

Below is the example provided for creating a container:



The screenshot shows the 'Processing' tab in the Azure Cosmos DB configuration tool. It is divided into two sections: 'PROCESSING DETAILS' and 'HEADERS DETAILS'. In the 'PROCESSING DETAILS' section, the 'Processing Mode' is set to 'Single', the 'Version' is '2018-12-31', the 'Entity' is 'Container', and the 'Operation' is 'Create'. The 'Database' field contains the placeholder text '\${header.DatabaseID}'. The 'HEADERS DETAILS' section has 'Request Headers' and 'Response Headers' fields, with the 'Response Headers' field containing an asterisk (\*).

Sample payload to create a container:



```

{
  "id": "testcoll",
  "indexingPolicy": {
    "automatic": true,
    "indexingMode": "Consistent",
    "includedPaths": [
      {
        "path": "/*",
        "indexes": [
          {
            "dataType": "String",
            "precision": -1,
            "kind": "Range"
          }
        ]
      }
    ]
  },
  "partitionKey": {
    "paths": [
      "/AccountNumber"
    ],
    "kind": "Hash",
    "Version": 2
  }
}

```

## 5.1.2 Delete

The **Delete** operation can be used to delete a Database, Container, or Document in Azure Cosmos DB.

You can set the Processing mode to **Single**, select the API Version, choose an Entity type from options such as **Database**, **Container**, and **Document**, and select operation as **Delete**. For more information about the configuration fields, see [Processing Tab](#).

Below is the example provided for deleting a database:

The screenshot shows the 'Processing' tab in the Azure Cosmos DB interface. Under 'PROCESSING DETAILS', the 'Processing Mode' is set to 'Single', 'Version' is '2018-12-31', 'Entity' is 'Database', and 'Operation' is 'Delete'. The 'Database' field contains the placeholder text '\$(header.DatabaseID)'. Under 'HEADERS DETAILS', the 'Request Headers' and 'Response Headers' fields are empty, with the 'Response Headers' field containing an asterisk '\*'.

### 5.1.3 Get

The **Get** operation can be used to fetch details about a Database, Container, or Document in Azure Cosmos DB.

You can set the Processing mode to **Single**, select the API Version, choose an **Entity** type from options such as **Database**, **Container**, and **Document**, and select operation as **Get**. For more information about the configuration fields, see [Processing Tab](#).

Below is the example provided to get a document:

The screenshot shows the 'Processing' tab in the Azure Cosmos DB interface. Under 'PROCESSING DETAILS', the 'Processing Mode' is set to 'Single', 'Version' is '2018-12-31', 'Entity' is 'Document', and 'Operation' is 'Get'. The 'Database' field contains '\$(header.DatabaseID)', 'Container' contains '\$(header.ContainerID)', and 'Document' contains '\$(header.DocumentID)'. The 'Partition Key' field is empty. Under 'HEADERS DETAILS', the 'Request Headers' and 'Response Headers' fields are empty, with the 'Response Headers' field containing an asterisk '\*'.

### 5.1.4 List

The **List** operation can be used to retrieve a list of resources such as Database, Container, or Document in Azure Cosmos DB.

You can set the Processing mode to **Single**, select the API Version, choose an **Entity** type from options such as **Database**, **Container**, and **Document**, and select operation as **List**. For more information about the configuration fields, see [Processing Tab](#).

Below is the example provided to List the Database:

The screenshot shows the 'Processing' tab in the Azure Cosmos DB interface. Under 'PROCESSING DETAILS', the 'Processing Mode' is set to 'Single', 'Version' is '2018-12-31', 'Entity' is 'Database', and 'Operation' is 'List'. Under 'HEADERS DETAILS', 'Request Headers' is empty and 'Response Headers' is set to '\*'.

## 5.1.5 Query

The **Query** operation can be used to query a Document in Azure Cosmos DB.

You can set the Processing mode to **Single**, select the API Version, choose an Entity type as **Document**, and select operation as **Query**. For more information about the configuration fields, see [Processing Tab](#).

Below is the example provided to Query a Document:

The screenshot shows the 'Processing' tab in the Azure Cosmos DB interface. Under 'PROCESSING DETAILS', the 'Processing Mode' is 'Single', 'Version' is '2018-12-31', 'Entity' is 'Document', and 'Operation' is 'Query'. The 'Database' field contains the expression '\$(header.DatabaseID)' and the 'Container' field contains '\$(header.ContainerID)'. Under 'HEADERS DETAILS', 'Request Headers' is empty and 'Response Headers' is set to '\*'.

Sample payload for Query documents:

```
{  "query": "SELECT * FROM Families f WHERE f.id = @id AND f.Address.City = @city",  "parameters": [    {      "name": "@id",      "value": "AndersenFamily"    },    {      "name": "@city",      "value": "Seattle"    }  ]}
```

## 5.1.6 Update

The **Update** operation can be used to modify the contents of an existing Document within a container in Azure Cosmos DB.

You can set the Processing mode to **Single**, select the API Version, choose an Entity type as **Document**, and select operation as **Update**. For more information about the configuration fields, see [Processing Tab](#).

Below is the example provided to update a Document:

The screenshot shows the 'Processing' tab in the Azure Cosmos DB management console. The interface is divided into two main sections: 'PROCESSING DETAILS' and 'HEADERS DETAILS'. In the 'PROCESSING DETAILS' section, the following settings are visible: 'Processing Mode' is set to 'Single'; 'Version' is set to '2018-12-31'; 'Entity' is set to 'Document'; 'Operation' is set to 'Update'; 'Database' is set to '\${header.DatabaseID}'; 'Container' is set to '\${header.ContainerID}'; 'Document' is set to '\${header.DocumentID}'; and 'Partition Key' is an empty field. The 'HEADERS DETAILS' section shows 'Request Headers' and 'Response Headers' as empty fields.

Sample payload:

```
{
  "operations": [
    {
      "op": "set",
      "path": "/Parents/0/FamilyName",
      "value": "Bob"
    }
  ]
}
```

## 5.1.7 Replace

The **Replace** operation can be used to completely overwrite an existing document with a new one for the Document entity and changes the indexing policy settings of a container after creation. Changing other properties of a container like the ID or the partition key is not supported.

You can set the Processing mode to **Single**, select the API Version, choose an Entity type from options such as **Container** or **Document**, and select operation as **Replace**. For more information about the configuration fields, see [Processing Tab](#).

Below is the example provided to replace a Document:

The screenshot shows the 'Processing' tab in the Azure Cosmos DB management console. The 'PROCESSING DETAILS' section is active, showing the following configuration:

- Processing Mode: Single
- Version: 2018-12-31
- Entity: Document
- Operation: Replace
- Database: \${header.DatabaseID}
- Container: \${header.ContainerID}
- Document: \${header.DocumentID}
- Partition Key: (empty)

The 'HEADERS DETAILS' section shows:

- Request Headers: (empty)
- Response Headers: \*

Sample payload for Replace document:

```
{
  "id": "_SalesOrder5",
  "AccountNumber": "NewUser01",
  "PurchaseOrderNumber": "P018009186470",
  "OrderDate": "2016-03-29T02:03:07.3526153Z",
  "Total": 5.95,
  "_rid": "d9RzAJRFKgwEAAAAAAAAA==",
  "_self": "dbs/d9RzAA==/colls/d9RzAJRFKgw=/docs/d9RzAJRFKgwEAAAAAAAAA==/",
  "_etag": "\"0000de86-0000-0000-0000-56f9e25c0000\"",
  "_ts": 1459216988,
  "_attachments": "attachments/",
  "shippedDate": "2016-03-29T02:03:07.4680723Z",
  "foo": "bar"
}
```

## 5.1.8 Get Partition Key Ranges

The **Get Partition Key Ranges** operation can be used to retrieve information about the partition key ranges within a Container, providing insights into how data is distributed across partitions.

You can set the Processing mode to **Single**, select the API Version, choose an Entity type as **Container**, and select operation as **Get Partition Key Ranges**. For more information about the configuration fields, see [Processing Tab](#).

The screenshot shows the 'Processing' tab in the Azure Cosmos DB management console. The interface is divided into two main sections: 'PROCESSING DETAILS' and 'HEADERS DETAILS'. In the 'PROCESSING DETAILS' section, the following settings are visible: 'Processing Mode' is set to 'Single'; 'Version' is set to '2018-12-31'; 'Entity' is set to 'Container'; 'Operation' is set to 'Get P...'; 'Database' is set to '\$(header.DatabaseID)'; and 'Container' is set to '\$(header.ContainerID)'. The 'HEADERS DETAILS' section shows 'Request Headers' as an empty text box and 'Response Headers' as an asterisk (\*).

## 5.2 Bulk Processing Mode

The following operation for bulk processing pertains to the **Document** entity within Azure Cosmos DB.

### 5.2.1 Create


The **Create** operation can be used to create multiple Documents in a single batch request in Azure Cosmos DB.



The screenshot shows the 'Processing' tab in the Azure Cosmos DB configuration interface. It is titled 'AzureCosmosDB' and includes an 'Externalize' button and a help icon. The 'Processing' tab is active, showing 'PROCESSING DETAILS' with the following fields: 'Processing Mode' set to 'Bulk', 'Entity' set to 'Document', and 'Operation' set to 'Create'. Below these are three input fields: 'Database: \*' with the value '\${header.DatabaseID}', 'Container: \*' with the value '\${header.ContainerID}', and 'Partition Key Path: \*' with the value '\${header.PartitionKeyPath}'. At the bottom, the 'HEADERS DETAILS' section shows 'Response Headers' set to '\*'. The interface also has 'General' and 'Connection' tabs.

For more information about the configuration fields, see [Processing Tab](#).

Sample payload:

 Note that the following information must be present in the payload which denotes the Document data.

```
[
  {
    "id": "_id",
    "partitionKey": "_partitionKey"
  },
  {
    "id": "_id",
    "partitionKey": "_partitionKey"
  }
]
```

### 5.2.2 Replace

The **Replace** operation in Azure Cosmos DB involves replacing/updating multiple documents in a single batch operation. This is useful when you need to modify existing documents in bulk rather than inserting new ones.

For more information about the configuration fields, see [Processing Tab](#).  
 Sample payload:

**i** Note that the following information must be present in the payload which denotes the Document data.

```
[
  {
    "id": "_id",
    "partitionKey": "_partitionKey"
  },
  {
    "id": "_id",
    "partitionKey": "_partitionKey"
  }
]
```

### 5.2.3Upsert

The **Upsert** operation is used to insert or update multiple documents in a single operation. It updates any existing documents in CosmoDB and insert it if they are not already present.



For more information about the configuration fields, see [Processing Tab](#).

Sample payload:



Note that the following information must be present in the payload which denotes the Document data.

```
[
  {
    "id": "_id",
    "partitionKey": "_partitionKey"
  },
  {
    "id": "_id",
    "partitionKey": "_partitionKey"
  }
]
```

## 5.2.4 Delete

A **Delete** operation refers to the process of deleting multiple documents in a single operation.

The screenshot shows the 'Processing' tab in the Azure Cosmos DB interface. It is divided into two sections: 'PROCESSING DETAILS' and 'HEADERS DETAILS'. In the 'PROCESSING DETAILS' section, the 'Processing Mode' is set to 'Bulk', the 'Entity' is 'Document', and the 'Operation' is 'Delete'. The 'Database', 'Container', and 'Partition Key Path' fields are all populated with the placeholder text `\${header.DatabaseID}`, `\${header.ContainerID}`, and `\${header.PartitionKeyPath}` respectively. The 'HEADERS DETAILS' section shows the 'Response Headers' field set to an asterisk (\*).

For more information about the configuration fields, see [Processing Tab](#).

Sample payload:

```
[
  {
    "id": "_id",
    "partitionKey": "_partitionKey"
  },
  {
    "id": "_id",
    "partitionKey": "_partitionKey"
  }
]
```

## 5.2.5 Patch

A **Patch** operation involves path-level updates to specific files/properties to multiple documents in a single batch. These operations include modifications such as add, increment, move, remove, replace, and set.

Patch operations are particularly useful for making partial updates to documents without needing to replace the entire document. You can choose the appropriate Patch Operations from the table below. For more information about the configuration fields, see [Processing Tab](#).

The screenshot shows the 'Processing' tab in the Azure Cosmos DB interface. It features a 'PROCESSING DETAILS' section with several configuration fields:

- Processing Mode: Bulk (dropdown)
- Entity: Document (dropdown)
- Operation: Patch (dropdown)
- Patch Operation: Add (dropdown)
- Database: \* \${header.DatabaseID} (text input)
- Container: \* \${header.ContainerID} (text input)
- Partition Key Path: \* \${header.PartitionKeyPath} (text input)
- Path: \* \${header.Path} (text input)

Patch Operation type	Description
<b>Add</b>	Add a new element or value to an existing document.
<b>Set</b>	Set or update the value of an element.
<b>Replace</b>	Replace operation is similar to Set except it follows <i>strict</i> replace-only semantics.
<b>Remove</b>	Remove operation deletes the element specified in the target path.
<b>Increment</b>	Increment a field by the specified value. It can accept both positive and negative values.
<b>Move</b>	Move operation removes the value at a specified location and adds it to the target location.

For more information about the above fields, see [Azure Cosmos DB documentation](#).

Sample payload:

```
[
  {
    "id": "_id",
    "partitionKey": "_partitionKey",
    "patchValue": "_patchValue"
  },
  {
    "id": "_id",
    "partitionKey": "_partitionKey",
    "patchValue": "_patchValue"
  }
]
```

Below is the sample response body provided for the bulk operations:

```
{
  "totalDocuments": 2,
  "successDocuments": 2,
  "failedDocuments": 0,
  "responses": [
    {
      "id": "1",
      "message": "The operation for Item ID [1] completed successfully with a 201 response code.",
      "statusCode": 201,
      "requestCharge": 14.666666666666666
    },
    {
      "id": "2",
      "message": "The operation for Item ID [2] completed successfully with a 201 response code.",
      "statusCode": 201,
      "requestCharge": 14.666666666666666
    }
  ]
}
```