**Cosmos DB API Management**: We need to do to get data, update date, delete data, create data using API in API management services.

A diagram of a diagram

Description automatically generated

**Add item API** : It is use to create item in Cosmos dB using create api. In inbound processing section you can able to see the code window icon, click on that you can able to see the code for creating item.

 <set-variable name="requestBody" value="@(context.Request.Body.As<JObject>( preserveContent:true ))" />

        <set-variable name="extractedValue" value="@((string)((JObject)context.Variables["requestBody"])["category"])" />

        <set-backend-service base-url="https://edmsand-cosmosdb-account.documents.azure.com:443/" />

        <set-method>POST</set-method>

        <authentication-managed-identity resource="https://edmsand-cosmosdb-account.documents.azure.com" output-token-variable-name="msi-access-token" ignore-error="false" />

        <set-header name="Authorization" exists-action="override">

            <value>@("type=aad&ver=1.0&sig=" + context.Variables["msi-access-token"])</value>

        </set-header>

        <set-header name="x-ms-date" exists-action="override">

            <value>@(DateTime.UtcNow.ToString("r"))</value>

        </set-header>

        <set-header name="x-ms-version" exists-action="override">

            <value>2018-12-31</value>

        </set-header>

        <set-header name="Content-Type" exists-action="override">

            <value>application/json</value>

        </set-header>

        <set-header name="x-ms-documentdb-partitionkey" exists-action="override">

            <value>@("[\"" + context.Variables["extractedValue"] + "\"]")</value>

        </set-header>

        <rewrite-uri template="/dbs/POC/colls/CountryAndFruit/docs" copy-unmatched-params="false" />

The above code we need to past in inside <inbound> tag.

**Example json code for creating item**:

{"Category":"Fruit","id":"4","foo":"bar"}

**Delete item API**: It is used to delete item in Cosmos dB using delete API and we need to pass the **id** as parameter. In inbound processing section you can be able to see the code window icon, click on that you can be able to see the code for creating item.

<base />

        <set-variable name="requestBody" value="@(context.Request.Body.As<JObject>( preserveContent:true ))" />

        <set-variable name="extractedValue" value="@((string)((JObject)context.Variables["requestBody"])["category"])" />

        <set-variable name="id" value="@(context.Request.MatchedParameters["id"])" />

        <!-- Check if id is not provided, then throw an error -->

        <choose>

            <when condition="@(!context.Variables.ContainsKey("id") || string.IsNullOrEmpty((string)context.Variables["id"]))">

                <return-response>

                    <set-status code="400" reason="Bad Request" />

                    <set-body>

<value>{"error": "The 'id' parameter is required."}</value></set-body>

                </return-response>

            </when>

            <otherwise>

                <!-- Continue with normal processing if id is present -->

                <set-backend-service base-url="https://edmsand-cosmosdb-account.documents.azure.com:443/" />

                <set-method>DELETE</set-method>

                <authentication-managed-identity resource="https://edmsand-cosmosdb-account.documents.azure.com" output-token-variable-name="msi-access-token" ignore-error="false" />

                <set-header name="Authorization" exists-action="override">

                    <value>@("type=aad&ver=1.0&sig=" + context.Variables["msi-access-token"])</value>

                </set-header>

                <set-header name="x-ms-date" exists-action="override">

                    <value>@(DateTime.UtcNow.ToString("r"))</value>

                </set-header>

                <set-header name="x-ms-version" exists-action="override">

                    <value>2018-12-31</value>

                </set-header>

                <set-header name="x-ms-documentdb-partitionkey" exists-action="override">

                    <value>@("[\"" + context.Variables["extractedValue"] + "\"]")</value>

                </set-header>

                <rewrite-uri template="/dbs/POC/colls/CountryAndFruit/docs/{id}" copy-unmatched-params="false" />

            </otherwise>

        </choose>

The above code we need to past in inside <inbound> tag.

**Example json code for Delete item**:

{"category":"Fruit","id":"24","foo":"bar"}

**Update item API**: It is used to update item in Cosmos dB using update item API and we need to pass the **id** as parameter. In inbound processing section you can be able to see the code window icon, click on that you can be able to see the code for creating item.

 <base />

        <set-variable name="requestBody" value="@(context.Request.Body.As<JObject>( preserveContent:true ))" />

        <set-variable name="extractedValue" value="@((string)((JObject)context.Variables["requestBody"])["category"])" />

        <set-variable name="id" value="@(context.Request.MatchedParameters["id"])" />

        <!-- Check if id is not provided, then throw an error -->

        <choose>

            <when condition="@(!context.Variables.ContainsKey("id") || string.IsNullOrEmpty((string)context.Variables["id"]))">

                <return-response>

                    <set-status code="400" reason="Bad Request" />

                    <set-body>

<value>{"error": "The 'id' parameter is required."}</value></set-body>

                </return-response>

            </when>

            <otherwise>

                <!-- Continue with normal processing if id is present -->

                <set-backend-service base-url="https://edmsand-cosmosdb-account.documents.azure.com:443/" />

                <set-method>PUT</set-method>

                <authentication-managed-identity resource="https://edmsand-cosmosdb-account.documents.azure.com" output-token-variable-name="msi-access-token" ignore-error="false" />

                <set-header name="Authorization" exists-action="override">

                    <value>@("type=aad&ver=1.0&sig=" + context.Variables["msi-access-token"])</value>

                </set-header>

                <set-header name="x-ms-date" exists-action="override">

                    <value>@(DateTime.UtcNow.ToString("r"))</value>

                </set-header>

                <set-header name="x-ms-version" exists-action="override">

                    <value>2018-12-31</value>

                </set-header>

                <set-header name="x-ms-documentdb-partitionkey" exists-action="override">

                    <value>@("[\"" + context.Variables["extractedValue"] + "\"]")</value>

                </set-header>

                <rewrite-uri template="/dbs/POC/colls/CountryAndFruit/docs/{id}" copy-unmatched-params="false" />

            </otherwise>

        </choose>

The above code we need to past in inside <inbound> tag.

**Example json code for Update item**:

{"category":"Fruit","id":"24","foo":"bar"}

**Get item API**: It is used to get item in Cosmos dB using get item API and we need to pass the **id** as parameter. In inbound processing section you can be able to see the code window icon, click on that you can be able to see the code for creating item.

<base />

        <!-- Extract the id from the URL path and store it in a variable -->

        <set-variable name="id" value="@(context.Request.MatchedParameters["id"])" />

        <!-- Check if id is not provided, then throw an error -->

        <choose>

            <when condition="@(!context.Variables.ContainsKey("id") || string.IsNullOrEmpty((string)context.Variables["id"]))">

                <return-response>

                    <set-status code="400" reason="Bad Request" />

                    <set-body>

<value>{"error": "The 'id' parameter is required."}</value></set-body>

                </return-response>

            </when>

            <otherwise>

                <!-- Continue with normal processing if id is present -->

                <set-backend-service base-url="https://edmsand-cosmosdb-account.documents.azure.com:443/" />

                <set-method>POST</set-method>

                <authentication-managed-identity resource="https://edmsand-cosmosdb-account.documents.azure.com" output-token-variable-name="msi-access-token" ignore-error="false" />

                <set-header name="Authorization" exists-action="override">

                    <value>@("type=aad&ver=1.0&sig=" + context.Variables["msi-access-token"])</value>

                </set-header>

                <set-header name="x-ms-date" exists-action="override">

                    <value>@(DateTime.UtcNow.ToString("r"))</value>

                </set-header>

                <set-header name="x-ms-version" exists-action="override">

                    <value>2018-12-31</value>

                </set-header>

                <set-header name="x-ms-documentdb-isquery" exists-action="override">

                    <value>True</value>

                </set-header>

                <set-header name="x-ms-documentdb-query-enablecrosspartition" exists-action="override">

                    <value>True</value>

                </set-header>

                <set-header name="Content-Type" exists-action="override">

                    <value>application/query+json</value>

                </set-header>

                <rewrite-uri template="/dbs/POC/colls/CountryAndFruit/docs" copy-unmatched-params="false" />

                <!-- Construct the JSON body dynamically using the extracted id variable -->

                <set-body>@("{\"query\":\"SELECT \* FROM  c WHERE c.id = @id\",\"parameters\":[{\"name\":\"@id\",\"value\":\"" + context.Variables["id"] + "\"}]}")</set-body>

            </otherwise>

        </choose>

The above code we need to past in inside <inbound> tag.

**Example json code for Get item by Id**:

{"category":"Fruit","id":"24","foo":"bar"}

**Get item by Partition Key API**: It is used to get item by partition key in Cosmos dB using get item by partition key API and we need to pass the category as parameter. In inbound processing section you can be able to see the code window icon, click on that you can be able to see the code for creating item.

 <base />

        <set-backend-service base-url="https://edmsand-cosmosdb-account.documents.azure.com:443/" />

        <authentication-managed-identity resource="https://edmsand-cosmosdb-account.documents.azure.com" output-token-variable-name="msi-access-token" ignore-error="false" />

        <set-header name="Authorization" exists-action="override">

            <value>@("type=aad&ver=1.0&sig=" + context.Variables["msi-access-token"])</value>

        </set-header>

        <set-header name="x-ms-date" exists-action="override">

            <value>@(DateTime.UtcNow.ToString("r"))</value>

        </set-header>

        <set-header name="x-ms-version" exists-action="override">

            <value>2018-12-31</value>

        </set-header>

        <rewrite-uri template="/dbs/POC/colls/CountryAndFruit/docs" copy-unmatched-params="false" />

        <set-header name="x-ms-documentdb-partitionkey" exists-action="override">

            <value>@("[\"" + context.Request.MatchedParameters["category"] + "\"]")</value>

        </set-header>

The above code we need to past in inside <inbound> tag

**Example json code for Get item by partition key**: on this case we need to pass the category field value as parameter.

{"category":"Fruit","id":"24","foo":"bar"}

**Patch item by id API**: It is used to patching item in Cosmos dB using patch item API and we need to pass the **id** as parameter. And if you need to pass any property and value you can pass. In inbound processing section you can be able to see the code window icon, click on that you can be able to see the code for creating item.

 <base />

        <set-variable name="requestBody" value="@(context.Request.Body.As<JObject>( preserveContent:true ))" />

        <set-variable name="extractedValue" value="@((string)((JObject)context.Variables["requestBody"])["category"])" />

        <set-variable name="id" value="@(context.Request.MatchedParameters["id"])" />

        <choose>

            <when condition="@(!context.Variables.ContainsKey("id") || string.IsNullOrEmpty((string)context.Variables["id"]))">

                <return-response>

                    <set-status code="400" reason="Bad Request" />

                    <set-body>

<value>{"error": "The 'id' parameter is required."}</value></set-body>

                </return-response>

            </when>

            <otherwise>

                <set-backend-service base-url="https://edmsand-cosmosdb-account.documents.azure.com:443/" />

                <set-method>PATCH</set-method>

                <authentication-managed-identity resource="https://edmsand-cosmosdb-account.documents.azure.com" output-token-variable-name="msi-access-token" ignore-error="false" />

                <set-header name="Authorization" exists-action="override">

                    <value>@("type=aad&ver=1.0&sig=" + context.Variables["msi-access-token"])</value>

                </set-header>

                <set-header name="x-ms-date" exists-action="override">

                    <value>@(DateTime.UtcNow.ToString("r"))</value>

                </set-header>

                <set-header name="x-ms-version" exists-action="override">

                    <value>2018-12-31</value>

                </set-header>

                <set-header name="Content-Type" exists-action="override">

                    <value>application/json\_patch+json</value>

                </set-header>

                <set-header name="x-ms-documentdb-partitionkey" exists-action="override">

                    <value>@("[\"" + context.Variables["extractedValue"] + "\"]")</value>

                </set-header>

                <rewrite-uri template="/dbs/POC/colls/CountryAndFruit/docs/{id}" copy-unmatched-params="false" />

            </otherwise>

        </choose>

The above code we need to past in inside <inbound> tag.

**Example json code for patch item by partition key**: on this case we need to pass the category field value as parameter.

{"category":"Fruit","id":"24","foo":"bar"}

**API Testing in Post-man :**

Go to API management services instance -> Choose Api tab -> go to add item API.

In the add item Api tab we can able to see the option below like this.

A screenshot of a computer

Description automatically generated

Go to test tab we can test the api in same window or we can test the api in post man also.If we want to test the api in post-man take the Request url, Request body past the post man. Below Screen shot for Creating item in cosmosdb.A screenshot of a computer

Description automatically generated

**Get item By id**

Go to API management services instance -> Choose Api tab -> go to get item id API.

In the get item id Api tab we can able to see the option below like this.

A screenshot of a computer

Description automatically generated

Go to test tab we can test the api in same window or we can test the api in post man also.If we want to test the api in post-man take the Request url, Request body past the post man. Below Screen shot for get item by id in cosmosdb.

A screenshot of a computer

Description automatically generated

**Delete item By id**

Go to API management services instance -> Choose Api tab -> go to delete item id API.

In the delete item id Api tab we can able to see the option below like this.

A screenshot of a computer

Description automatically generated

Go to test tab we can test the api in same window or we can test the api in post man also. If we want to test the api in post-man take the Request url, Request body past the post man. Below Screen shot for get item by id in cosmosdb.

A screenshot of a computer

Description automatically generated

**Update item By id**

Go to API management services instance -> Choose Api tab -> go to delete item id API.

In the update item id api tab we can able to see the option below like this.

A screenshot of a computer

Description automatically generated

Go to test tab we can test the api in same window or we can test the api in post man also. If we want to test the api in post-man take the Request url, Request body past the post man. Below Screen shot for update item by id in cosmosdb.

A screenshot of a computer

Description automatically generated

**Creating api in API Management Services:**

1. Click on api tab
2. Click on add operation.
3. Go to design tab and give the displayname, Name, url.

A screenshot of a computer

Description automatically generated

1. Click on save. It will create api.
2. Go to Settings tab if you want to edit the name and url do it. A screenshot of a computer

   Description automatically generated
3. Click on Test tab, if we want to test the api. If we want to pass the parameter and header we can pass it here.A screenshot of a computer

   Description automatically generated
4. After creating api then we can able to see the api tab below here.

A screenshot of a computer

Description automatically generated

1. On the inbond policy rules section, we need to specify the rules code. click on that we need to see the code. A screenshot of a computer

   Description automatically generated
2. On back end section we need to mention the cosmosdb api instance url. ssA screenshot of a computer

   Description automatically generated

**Azure Pipelines :-**