**Logic App: Get Customer**

1. Create a logic App Instance from Azure portal.
2. Navigate to Logic App Designer.
3. Select “When a Http Request is received” trigger from the list of common triggers.
4. In the JSON Body schema, enter information regarding the parameter that needs to be taken as input from the user. In case of no parameters from users, it can be left blank but for our scenario this needs to be entered.

{

"properties": {

"CustomerId": {

"type": "integer"

}

},

"type": "object"

}

1. Now, add a Control Action, and select condition from the list. Specify an “and” condition on the parameter “CustomerId” which was taken by user, i.e.

**CustomerId**(This can be taken from Dynamic Content)

**isequalto**(can be selected from multiple operations provided in the dropdown)

**Null**(This can be added as an expression)

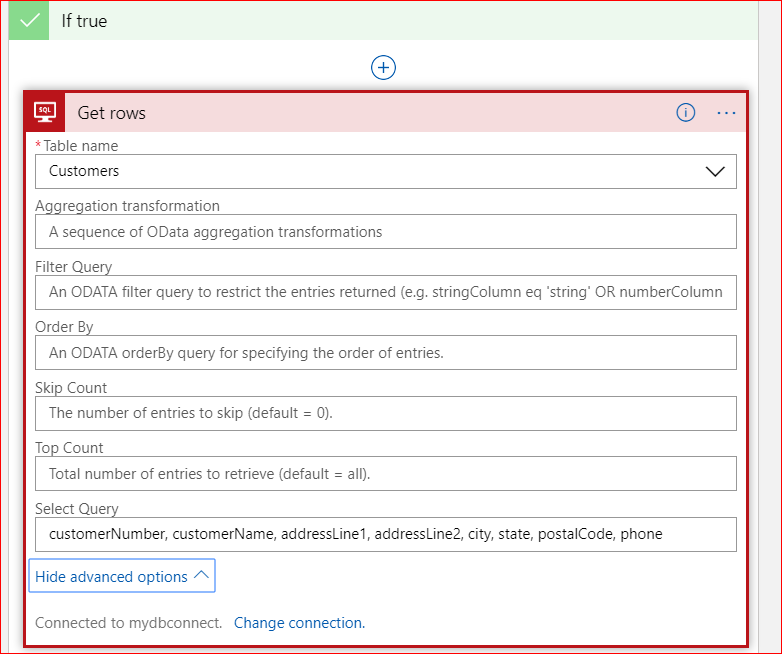
1. If condition is True:
   1. In the search tool, search for SQL Server and then Select “Get Rows” action under it.
   2. Specify the SQL Server information, like, connection name, SQL Server Name (If SQL server exists in the same resource group then can select the SQL Server from the list and proceed further, otherwise custom values can also be provided), Username and Password.

This will set-up an API connection which will be added as a resource in the resource group.

* 1. Now to get the rows as per requirements, select the table name and click advanced options:

As in this case, if condition is true, all the data needs to be returned, hence no condition will be applied, and only certain columns will be selected.

Requested columns: customerNumber, customerName, addressLine1, addressLine2, city, state, postalCode, phone

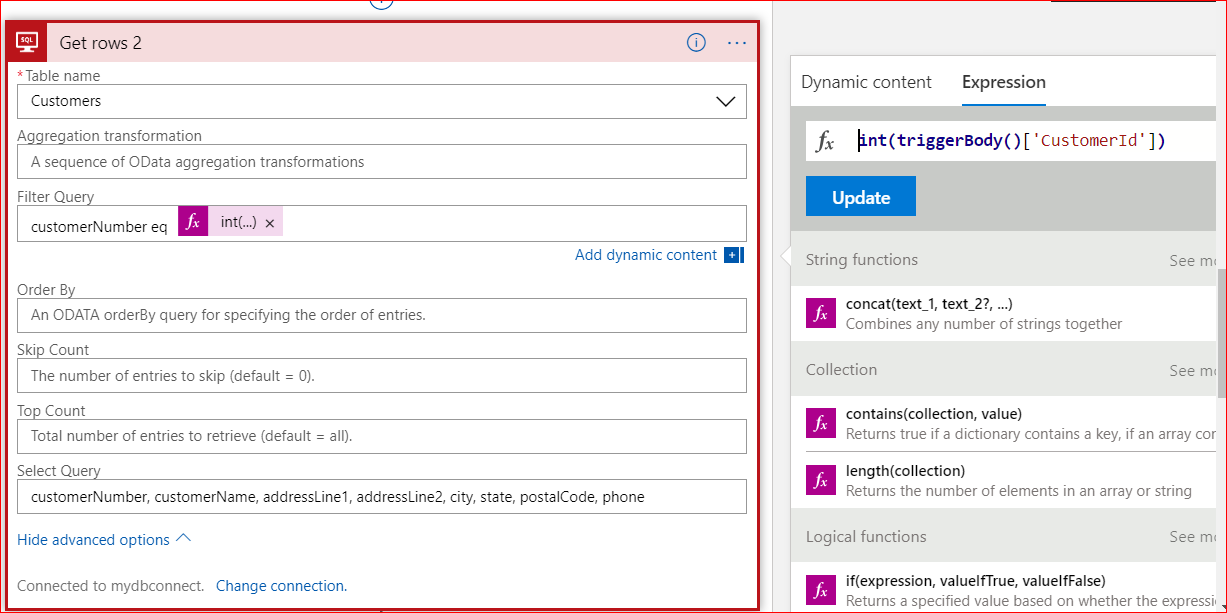


1. If condition is False:
   1. In the search tool, search for SQL Server and then Select “Get Rows” action under it.
   2. As the API-Connection is already been set-up, hence no need to configure it again.
   3. Now to get the rows as per requirements, select the table name and click advanced options:

As in this case, if condition is false, data specific to provided CustomerId needs to be returned, hence a condition will be applied, and only certain columns will be selected.

Condition: customerNumber eq **int(triggerBody()['CustomerId'])**(This will be added as an expression).

Requested columns: customerNumber, customerName, addressLine1, addressLine2, city, state, postalCode, phone



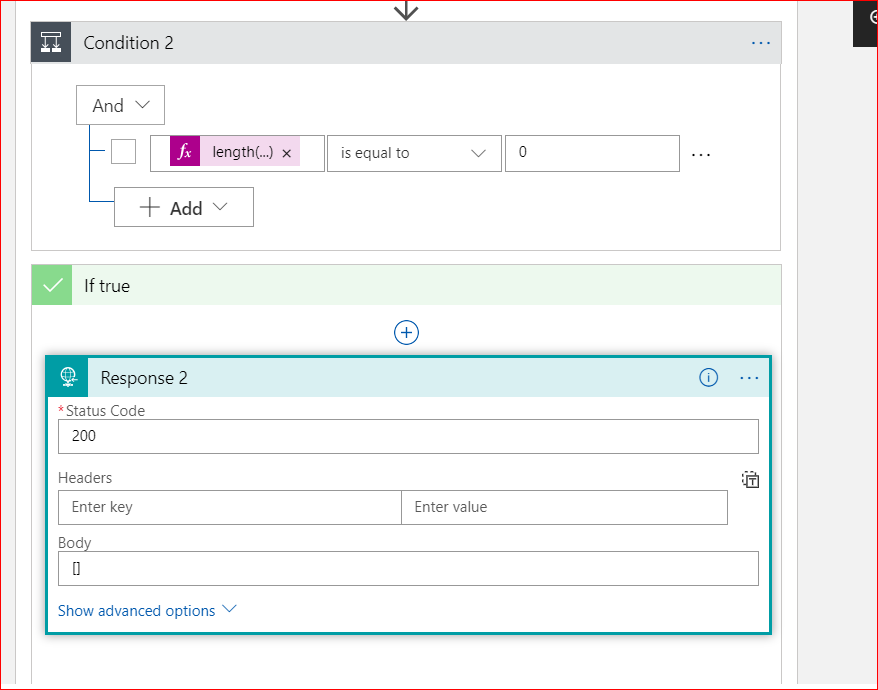
* 1. Now again a condition will be applied, add a Control Action, and select condition from the list. Specify an “and” condition on the length of response received. If number of rows returned is zero, then response will be an empty array.

**length(body('Get\_rows\_2')?['value'])**(This can be imported as an expression)

**isequalto**(can be selected from multiple operations provided in the dropdown)

**0**(can be written in the textbox).

1. If Condition2 is true:
   1. Select Response from the action items.
   2. Then a 200 response is returned as an empty array.



ii. If Condition is False: No need to configure any action.

1. Response:
   1. Select the action as Response.
   2. Now return a response with 200 status code and in body, select the values returned from both “Get Rows” operation.

