

Crear una consulta

Create Expression Rule

Create from scratch Duplicate existing expression rule

Name *

AX_GetVehicleByID

Description

Return a vehicle by its unique ID number

Save In *

AX Expressions x



Create New Rule Folder

CANCEL

CREATE

Query Editor

Select Data Source

Record Type Data Store Entity

Data Store Entity

AX_VEHICLE_DSE x

Data Preview

vehicleid	vehiclemake	vehiclemodel	vehiclecolor	vehiclecondition	vehiclestatus	vehiclecategory	vehiclemileage	vehicleyear	vehiclevin
1	Ford	F150	Red	10	9	3	500	2019	2F2DE48C8N4309374
2	Lexus	ES350	Pearl	11	9	6	7000	2019	2L3ED45V3D4030403

CANCEL

CONTINUE

The screenshot shows the 'Query Editor - AX VEHICLE DSE' interface. A modal dialog titled 'Rule Inputs and Test Values' is open, displaying a table with columns: Name, Type, Array, Expression, and Value. The first row contains 'vehicleid', 'Number (Integer)', a checked checkbox, '1', and an empty input field. Below the table is an 'Add New Rule Input' button. The background interface includes a left sidebar with 'Paging' and 'Filters' sections, and a bottom bar with 'CHANGE DATA STORE ENTITY', 'CANCEL', and 'GENERATE QUERY' buttons.

Query Editor - AX_VEHICLE_DSE

[RULE INPUTS](#)

▼ Paging & Sorting

Basic Expression

Show All Rows 1 Rows Per Page

Sort By Select a field In Ascending Order ×

▼ Filters

Basic Expression

Ignore filters with empty values

Only records that meet all filter conditions will be returned. [Learn more](#) about creating filters.

Field	Condition	Value	Apply Filter
<input checked="" type="checkbox"/> vehicleid ×	<	rivehicleid - Number (Integer)	Always ↑ ↓ ×

[Add Filter](#)

Fields

Query Results Preview

[ADD ALL](#) [REMOVE ALL](#) Fetch total row count

[GROUP DATA](#) [TEST QUERY](#)

[CHANGE DATA STORE ENTITY](#) [CANCEL](#)

[GENERATE QUERY](#)

AX_GetVehicleById

[SAVE](#)

[Ad Hoc Test](#) [Test Cases \(0\)](#)

```

1  alqueryEntity(
2    entity: cons!AX_VEHICLE_DSE,
3    query: alquery(
4      logicalExpression: alqueryLogicalExpression(
5        operator: "AND",
6        filters: {
7          alqueryFilter(
8            field: "vehicleid",
9            operator: "<",
10           value: rivehicleid
11          )
12        },
13        ignoreFiltersWithEmptyValues: true
14      ),
15      pagingInfo: alpagingInfo(
16        startIndex: 1,
17        batchSize: 1
18      )
19    ),
20    fetchTotalCount: false
21  ).data
    
```

Place cursor on function, rule, or constant to display help

▼ Test Inputs

Rule Input Name	Expression	Value
vehicleid (Number (Integer))	1	

[Save as Test Case](#)

[TEST RULE](#)

▼ Test Output

Time 123 ms [View Performance](#) Type List of Variant

Value Formatted Raw Expression

List of Dictionary - 1 item

- Dictionary
 - vehiclelastmodifieddate 10/7/2021 (Date)
 - vehiclelastmodifiedby "community.admin" (Text)
 - vehicledateadded 5/4/2019 (Date)
 - vehicleaddedby "regina.registrar" (Text)
 - vehicleimage 3544 (Number (Integer))
 - vehiclenextmaintenancedate 8/10/2020 (Date)
 - vehiclelastmaintenancedate 6/8/2020 (Date)
 - vehiclevin "2F2DE48C8N4309374" (Text)
 - vehicleyear 2019 (Number (Integer))
 - vehiclemileage 500 (Number (Integer))
 - vehiclecategory 3 (Number (Integer))
 - vehiclestatus 9 (Number (Integer))
 - vehiclecondition 10 (Number (Integer))
 - vehiclecolor "Red" (Text)