

Filter Action Define

Comparison Query Operators <op>

NAME	Operator
GREATER_THAN	gt
LESS_THAN	lt
GREATER_THAN_OR_EQUA	gte
LESS_THAN_OR_EQUA	lte
EQUA	eq
NOT_EQUA	neq
IN	in
NOT_IN	nin
LIKE	like

Logical Data Types <logicalDataType>

key	The field name
op	Comparison query operators
value	The input value

Select

1. Operator:

Case	Operators
Single	<ul style="list-style-type: none">EQUANOT_EQUA

2. Format:

Case	Format	Example

Single	<pre>{ [key]: { op: <op>, value: <Array> } }</pre>	<pre>{ status: { op: 'eq', value: ['public', 'private'] } }</pre>
---------------	--	---

Boolean

1. Operator:

Case	Operators
Single	<ul style="list-style-type: none"> • EQUA • NOT_EQUA
Multi	<ul style="list-style-type: none"> • IN • NOT_IN

2. Format:

Case	Format	Example
Single	<pre>{ [key]: { op: <op>, value: <Boolean> } }</pre>	<pre>{ status: { op: 'eq', value: true } }</pre>
Multiple	<pre>{ [key]: { op: <op>, value: <Array> } }</pre>	<pre>{ status: { op: 'in', value:[true, false] } }</pre>

Text

1. Operator:

Case	Operators
------	-----------

Single	<ul style="list-style-type: none"> • EQUA • NOT_EQUA • LIKE
Multi	<ul style="list-style-type: none"> • IN • NOT_IN • LIKE

2. Format

Case	Format	Example
Single	<pre>{ [key]: { op: <op>, value: <String> } }</pre>	<pre>{ name: { op: 'like', value: 'example text' } }</pre>
Multiple	<pre>{ [key]: { op: <op>, value: <Array> } }</pre>	<pre>{ name: { op: 'in', value: ['example text', 'example text 2'] } }</pre>

Number

1. Operator:

Case	Operators
Single	<ul style="list-style-type: none"> • GREATER_THAN • LESS_THAN • GREATER_THAN_OR_EQUA • LESS_THAN_OR_EQUA • EQUA • NOT_EQUA
Multi	<ul style="list-style-type: none"> • IN • NOT_IN

2. Format:

Case	Format	Example
Single	<pre>{ [key]: { op: <op>, value: <Number> } }</pre>	<pre>{ quanlity: { op: 'eq', value: 12 } }</pre>
Multiple	<pre>{ [key]: { op: <op>, value: <Array> } }</pre>	<pre>{ quanlity: { op: 'in', value: [13, 45] } }</pre>

Date

1. Operator:

Case	Operators
Single	<ul style="list-style-type: none">• GREATER_THAN• LESS_THAN• GREATER_THAN_OR_EQUA• LESS_THAN_OR_EQUA• EQUA• NOT_EQUA
Multi	<ul style="list-style-type: none">• IN• NOT_IN

2. Format:

Case	Format	Example
Single	<pre>{ [key]: { op: <op>, value: <Date> } }</pre>	<pre>{ createdAt: { op: 'eq', value: '2020-05-24T09:18:44.189Z' } }</pre>

	}	}
Multiple	<pre>{ [key]: { op: <op>, value: <Date> } }</pre>	<pre>{ createdAt: { op: 'in', value: ['2020-05-24T09:18:44.189Z', '2020-05-21T09:18:44.189Z'] } }</pre>

Date Range

1. Operator:

Case	Operators
Single	<ul style="list-style-type: none"> GREATER_THAN LESS_THAN GREATER_THAN_OR_EQUA LESS_THAN_OR_EQUA EQUA NOT_EQUA

2. Format:

Case	Format	Example
Single	<pre>{ [key]: { from: { op: <op>, value:<Date> }, to: { op: <op>, value: <Date> }, }, }</pre>	<pre>{ updatedAt: { from: { op: 'gt', value: '2020-05-24T09:18:44.189Z', }, to: { op: 'lt', value: '2020-05-27T09:18:44.189Z', }, }, }</pre>

Logical Query Operators <log>

NAME	Operator
AND	AND
OR	OR

Logical Query Data

Case	Format	Example
AND	<pre>{ <log>: [<logicalDataType>] }</pre>	<pre>{ AND: [{ quantity: { op: "eq", value: 12, }, }, { name: { op: "like", value: 'example text', }, },], }</pre>
OR	<pre>{ <log>: [<logicalDataType>] }</pre>	<pre>{ OR: [{ quantity: { op: "eq", value: 12,</pre>

		<pre> }, }, { name: { op: "like", value: 'example text', }, },], }</pre>
--	--	--