

Querying with OCNQL

Last Modified on 12/03/2020 5:52 pm EST

On this page

The Open Connectors Query Language (OCNQL) provides a standard way to search across all of our connectors. Many API providers support some form of searching in their APIs but each one approaches searching differently. Rather than having you research how to query each resource at a provider, we've normalized your search experience. SAP Cloud Platform Open Connectors translates your queries to the API provider's search syntax.

OCNQL Overview

OCNQL represent the "where" portion of a typical query that can include many other components like field selection, sorting, and pagination. Take a look at `GET` requests or `POST /bulk/query` in the API docs for any connector to see how you can structure your queries. Here are some examples from QuickBooks Online, MailChimp, and Salesforce Sales Cloud.

The image shows three API documentation snippets side-by-side, each with a red box highlighting the 'where' parameter. The snippets are for QuickBooks Online, MailChimp, and Salesforce Sales Cloud. Each snippet shows a table of parameters with 'where' highlighted. The 'where' parameter is described as a string representing a CEQL search expression or a WHERE clause in a typical SQL query. The QuickBooks Online snippet shows a GET request to /credit-memos. The MailChimp snippet shows a GET request to /lists. The Salesforce Sales Cloud snippet shows a GET request to /opportunities. The 'where' parameter is consistently defined across all three as a string representing a search expression.

The easiest way to make queries is through the API docs, but when you integrate your code with SAP Cloud Platform Open Connectors you'll need to understand how to structure queries with OCNQL.

You compose queries with available query connectors field selection, sorting, and pagination. Each connector allows different components of the query, but in most situations you can use some common parts. The most common components include the object name, OCNQL expression, field selection, field sorting, and pagination. Here's how you might construct a query:

