Data Fetcher Set Up and User Guide

Set Up and User Guide

Create. Innovate. Educate.

listing

Introduction

* Salesforce Labs provides free, open-source solutions to the AppExchange.

Data Fetcher allows admins and developers to dynamically query Salesforce records from within a single screen within flow utilizing a SOQL Query or SOSL.

System requirements:
Available for use in Enterprise, Unlimited, Force.com, Developer, and Performance Editions of Salesforce.

Data Fetcher is a free solution designed to optimize the user experience with the following Salesforce Products and thus requires

- Flow

Installation Guide

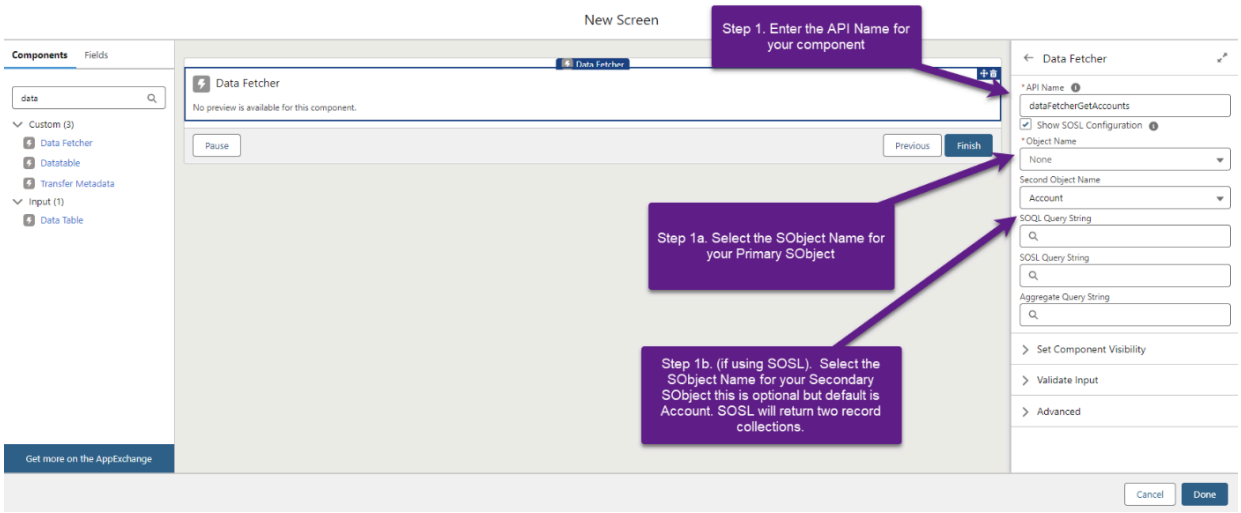
Login with your trailhead email and navigate to our listing on Appexchange

Select “Get it Now” from the listing and select the org you want to install it in from the drop-down

1. Agree to the terms and conditions
2. **Select “Get it Now” from the listing and select the org you want to install it in from the drop-down**
3. Agree to the terms and conditions
4. Once you have been redirected to the Salesforce login page, input the credentials for the Salesforce org you wish to install the solution in
5. Select the profiles you wish to access the solutions (Admins Only, Everyone, or Specific Profile)
6. The package includes several Apex Classes end users will need to be granted access to the class called DataFetcherController either via a permission set or profile. All other classes are only used during setup and configuration.

Set Up Guide

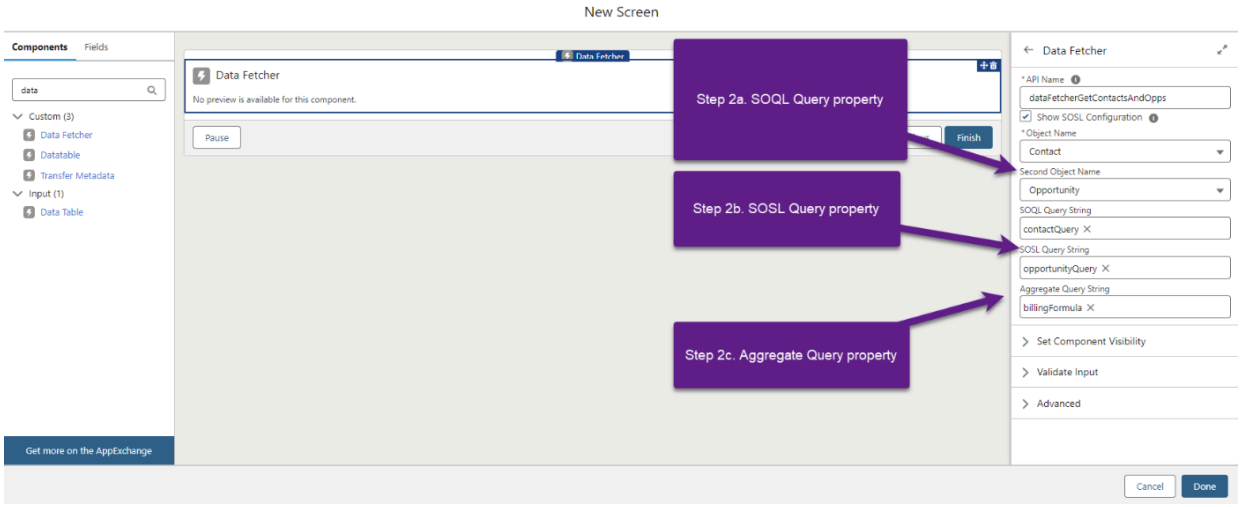
Once you have successfully installed Data Fetcher



1. Drag and drop Data Fetcher onto your flow screen this is a headless component so which means it will not show in the UI so you can place it where it makes sense to you on the screen and enter the API name you want to call Data Fetcher in the Flow.
    a. Next in the Object Name Property select the SObject you would like to query for either SOQL or SOSL configurations.
    b. If you are using SOSL you can set a secondary SObject that will return a second record collection. Select the Object Name for the SObject type you want to store in this collection.
2. Finally, enter in your query string/s when using Reactive Screens this can be an a formula, text template or can be a static

query.

a. SOQL Query is a standard SOQL and will return the first retrieved record and all records found in a record collection variable called retrievedRecords.
b. SOSL Query is a standard SOSL and will return two record collections in variables called searchResults and searchResults1 based off of the SObjects chosed in the first step.
c. Aggregate SOQL Query allows you to enter an Aggregate SOQL Query
d. Note: You can actually use all three query types at the same time but it is not recommended.



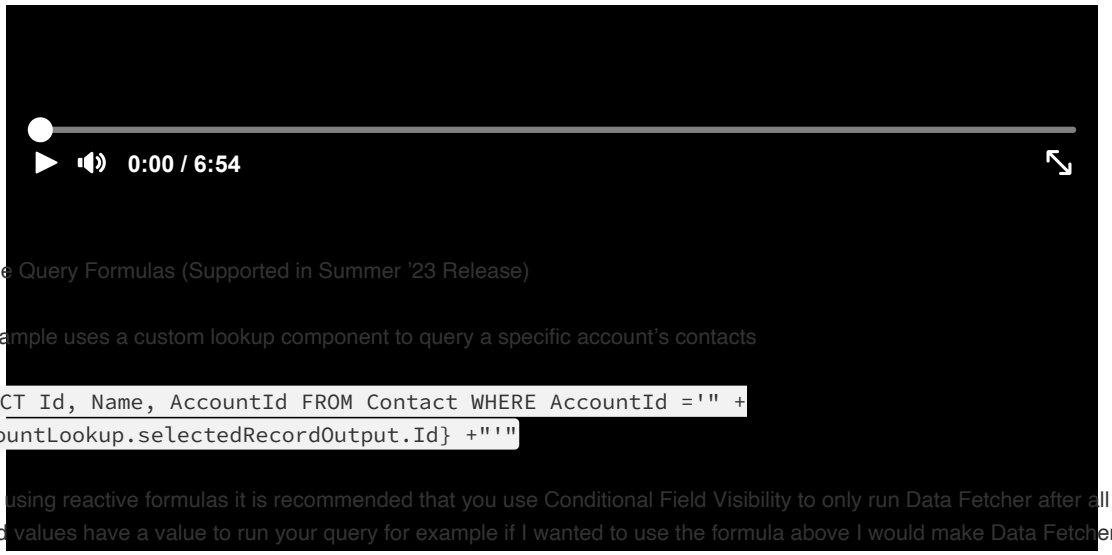
There are 4 Output variables for use on the same screen if Reactive Screens (Beta) is enabled in the org or for further use downstream in your flow.

1. aggQueryResult: This returns a number result from your Aggregate Query
2. Aggregate SOQL Query: This was the final query string for your Aggregate Query
3. error: Returns any errors that occurred from your query this also displays on the screen to the user in the event of an issue when querying records
4. firstRetreivedRecord: This is a record variable with the first record returned during your query
5. objectName1: This is the Object Name for the first SObject
6. objectName2: This is the Object Name for the second SObject
7. retrievedRecords: This is the record collection variable holding all of the records found.
8. searchResults: This is the first record collection from your SOSL Query
9. searchResults1: This is the second record collection from your SOSL Query
10. SOQL Query: This is the final query string for your SOQL Query
11. SOSL Query: This is the final query string for your SOSL Query


aggQueryResult
Search variables...
Aggregate SOQL Query
Search variables...
error
Search variables...
firstRetrievedRecord
Search variables...
objectName1
Search variables...
objectName2
Search variables...
retrievedRecords
Search variables...
searchResults
Search variables...
searchResults1
Search variables...
SOQL Query
Search variables...
SOSL Query String
Search variables...

Examples


0:00 / 6:40



Example Query Formulas (Supported in Summer '23 Release)

This example uses a custom lookup component to query a specific account's contacts

```
"SELECT Id, Name, AccountId FROM Contact WHERE AccountId ='" +
{!accountLookup.selectedRecordOutput.Id} +"'"
```

Note: If using reactive formulas it is recommended that you use Conditional Field Visibility to only run Data Fetcher after all required values have a value to run your query for example if I wanted to use the formula above I would make Data Fetcher condition using this formula '{!accountLookup.selectedRecordOutput.Id} isNull false' otherwise the user might see an error in the UI

This example is used for an Aggregate SOQL to return the total amount of queried Opportunities

```
"SELECT SUM(Amount) FROM Opportunity WHERE Account.BillingState = '" + {!Select_State} + "' AND
Account.Industry = '" + {!Select_Industry} + "'"
```

Resources

1. Follow Labs on Twitter!
2. Check out Labs on Github
3. Check out all Labs solutions on AppExchange
4. Trailblazer Community