

## Moving data out of Salesforce

Date published: 2019-06-26

Date modified: 2023-08-29

# CLouDERA

# Legal Notice

© Cloudera Inc. 2024. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

# Contents

<b>Moving data out of Salesforce using Apache NiFi.....</b>	<b>4</b>
Moving data out of Salesforce.....	4
Before you begin.....	4
Configuring Salesforce.....	4
Building your dataflow.....	7
Creating Controller Services for your dataflow.....	7
Configuring your source processor.....	9
Configuring your target processor.....	11
Confirming your dataflow success.....	12

## Moving data out of Salesforce using Apache NiFi

Cloudera supports moving data out of Salesforce using Apache NiFi. You can use Apache NiFi to move data out of Salesforce and push the data to other systems including Cloudera Data Platform.

### Moving data out of Salesforce

You can create a NiFi dataflow to move data out of Salesforce. To do this, you must meet some prerequisites, configure Salesforce, configure your controller services, build your dataflow, and configure the source and target processors.

Objects in Salesforce are database tables. The rows are known as records, and the columns are called fields. The `QuerySalesforceObject` processor queries Salesforce objects and retrieves their records.

The processor constructs the query using Salesforce Object Query Language (SOQL) and retrieves the result record dataset using the Salesforce REST API.

The processor utilizes streams and NiFi record-based processing to be able to handle a large number of records and to allow arbitrary output format.

### Before you begin

Before setting up a NiFi dataflow to pull data from Salesforce objects, you must meet certain minimum prerequisites.

- You have a Salesforce account.
- You have a Connected App running in Salesforce.
- You have a security token for authorization.

See the following sections for installation instructions appropriate for your deployment objectives:

- *Configuring Salesforce*
- *Creating Controller Services for your dataflow*

#### Related Information

[Configuring Salesforce](#)

[Creating Controller Services for your dataflow](#)

### Configuring Salesforce

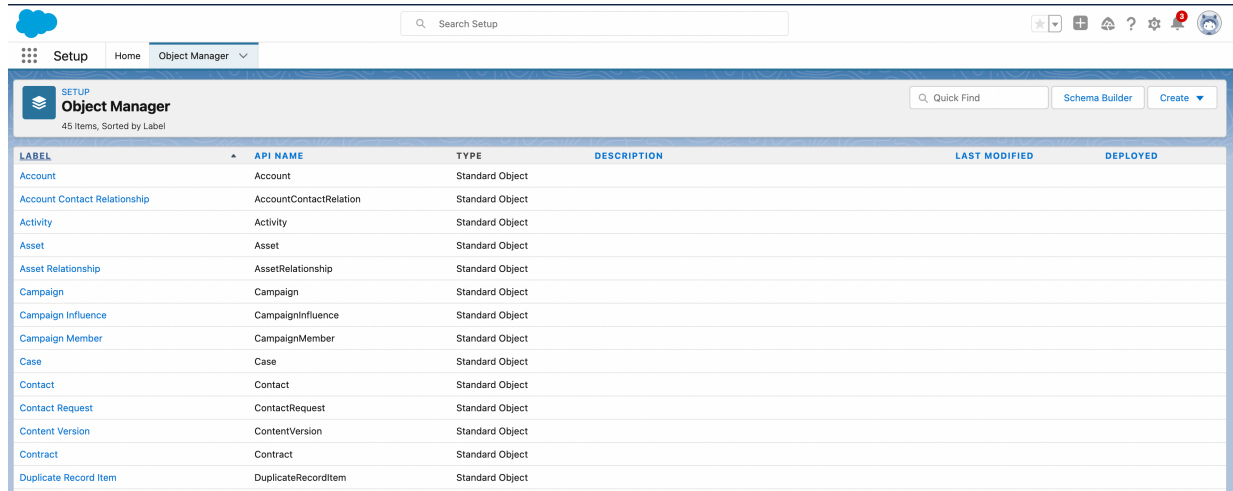
Learn how to configure Salesforce so that you can move data out of Salesforce by using Apache NiFi.

#### Procedure

1. Log in to Salesforce.
2. Click the gear icon at the top-right corner of the window.  
The Setup Home page appears.

3. Go to the Object Manager tab.

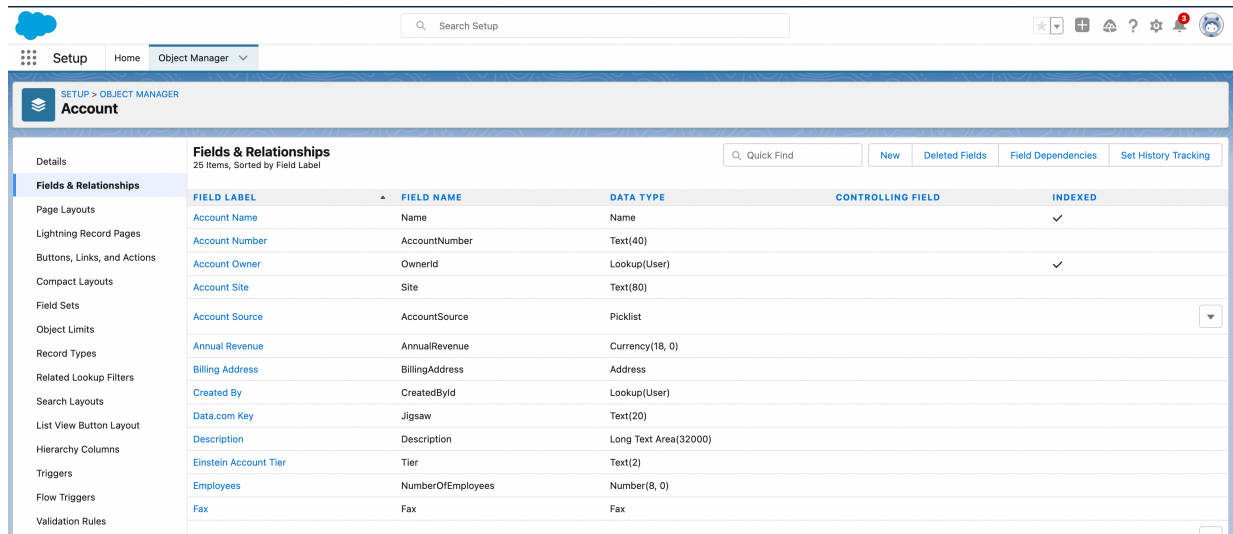
You can view the available standard Salesforce objects to be queried.



4. Select an object.

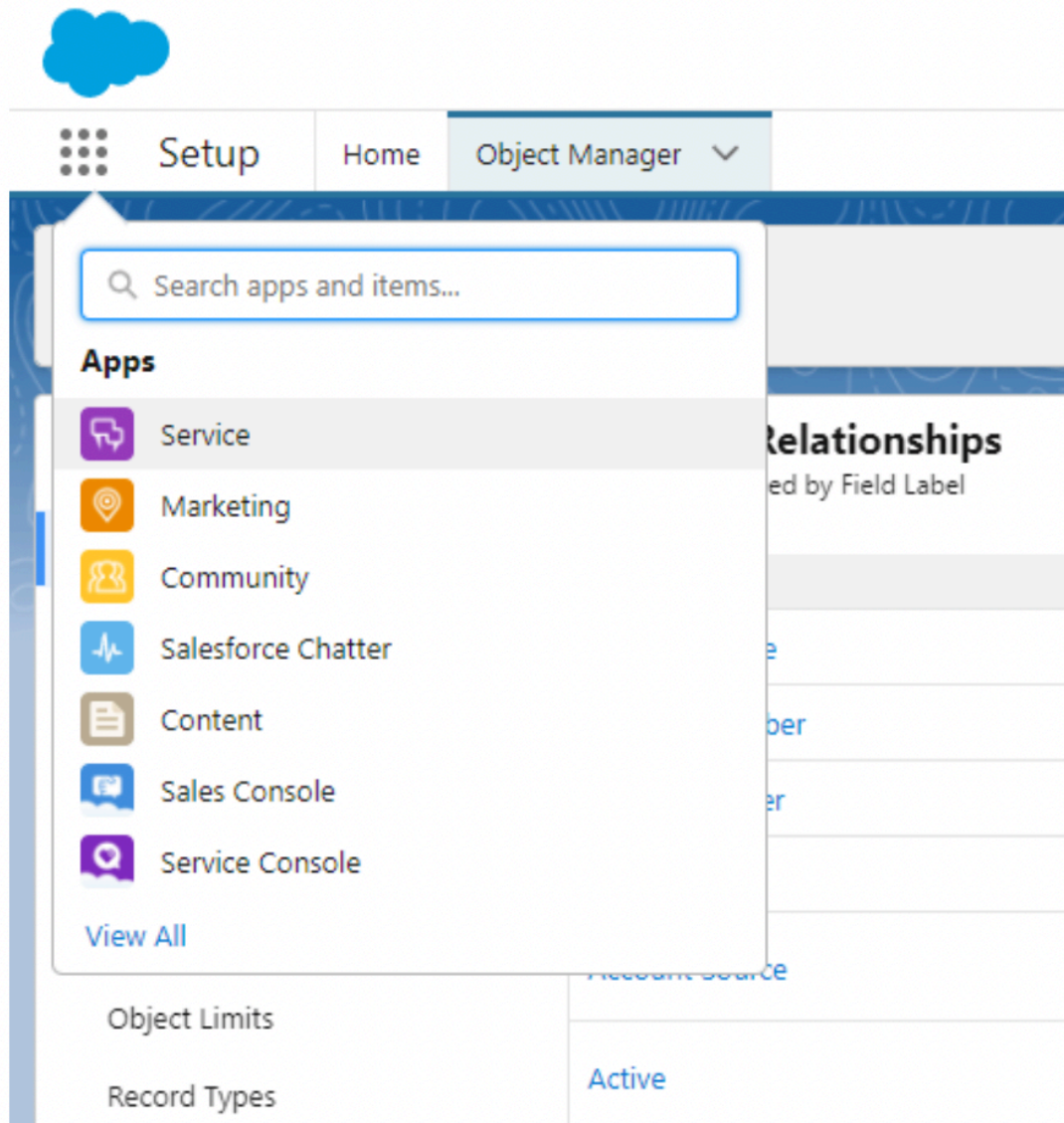
For example, select Account.

5. Go to the Fields & Relationships tab.

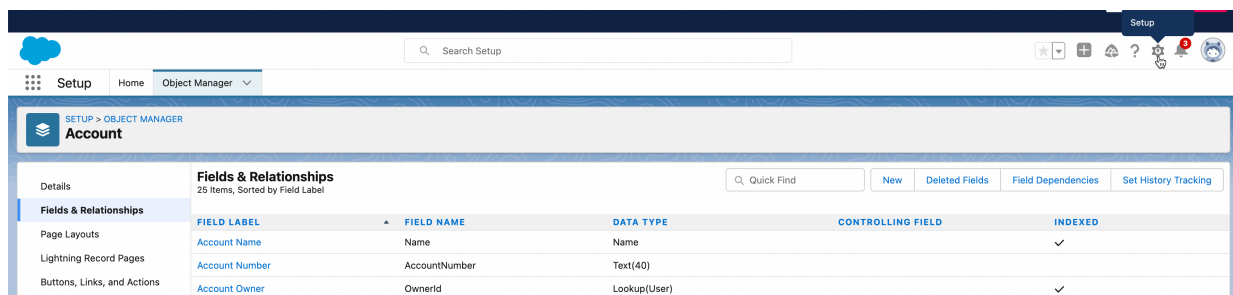


This tab displays some of the accessible fields (columns). These fields can be utilized in the Field Names processor property as well as the Custom WHERE condition. For more details, see *Salesforce Object Query Language (SOQL)*.

- Navigate to a Salesforce application, such as Service, to create a record for Account.



- Create a new account on the Service page's Account tab, which will be added as a record (row) to the Account table.
- Click the gear symbol in the top right corner of the window to return to the Setup page.



### Related Information

[Salesforce Object Query Language \(SOQL\)](#)

## Building your dataflow

Set up the elements of your NiFi dataflow that enables you to move data out of Salesforce using Apache NiFi. This involves opening NiFi in CDP Public Cloud or CDP Private Cloud Base cluster, adding processors to your NiFi canvas, and connecting the processors.

### About this task

When you are building a dataflow to move data out of Salesforce using Apache NiFi, you use the following source processor to build your dataflow:

- QuerySalesforceObject

### Procedure

1. Launch NiFi from your CDP Public Cloud or CDP Private Cloud Base cluster.
2. Add the NiFi processors to your canvas.
  - a) Select the Processor icon from the Cloudera Flow Management Actions pane, and drag a processor to the canvas.
  - b) Use the Add Processor filter box to search for the processor you want to add, and then click Add.
  - c) Add each of the processors you want to use for your dataflow.
3. Connect the two processors to create a data flow.
  - a) Click the Connection icon for the first processor, and drag it to the second processor.  
A Create Connection dialog displays. It has two tabs: Details and Settings, where you can configure the connection properties.
  - b) Click Add to close the dialog box and add the connection to your flow.  
Optionally, you can add success and failure funnels to your data flow, which help you to view where the flow files are routed when your dataflow is running.

### What to do next

After you have finished building the dataflow, move on to the following steps:

1. Create Controller Services for your dataflow
2. Configure your source processor
3. Configure your target processor

## Creating Controller Services for your dataflow

You can add controller services that can provide shared services to be used by the processors in your dataflow. Create them after you build the NiFi dataflow and before you configure the processors, so that they are available when you configure your NiFi processors.

### About this task

For this dataflow, create two controller services:

- StandardOAuth2AccessTokenProvider
- JsonRecordSetWriter

For complete details about these Apache NiFi controller services, see *NiFi Documentation*.

### Before you begin

You must have built your dataflow.

## Procedure

1. To add a controller service to your flow, right-click on the canvas and select Configure from the pop-up menu.  
The Controller Services Configuration window appears.
2. Select the Controller Services tab.
3. Click the + button to display the Add Controller Service dialog.
4. Select the required controller service and click Add.
5. Click the Configure icon in the right-hand column and configure the necessary options.
6. Click Apply to save the changes.
7. Click the Enable button (flash) in the far-right column of the Controller Services tab to enable the controller service.

## Example

You require the following controller services for this example of a NiFi dataflow moving data out of Salesforce.

StandardOAuth2AccessTokenProvider

This controller service allows NiFi to interact with the Salesforce database. The Salesforce REST API authorization is handled by the OAuth2 Access Token Provider Service.

**Controller Service Details** | StandardOAuth2AccessTokenProvider 1.17.0-SNAPSHOT

SETTINGS

PROPERTIES

COMMENTS

**Required field**

Property	Value
Authorization Server URL	<a href="https://cloudera-2f-dev-ed.my.salesforce.com/...">https://cloudera-2f-dev-ed.my.salesforce.com/...</a>
Grant Type	User Password
Username	lehelb@cloudera.com
Password	Sensitive value set
Client ID	3MVG9t0sl2P.pByqLVMOBOSvtIABv703Cqwom...
Client secret	Sensitive value set
Refresh Window	0 s
SSL Context Service	No value set

OK

You configure the following properties:

- Authorization Server URL
- Username
- Password
- Client ID
- Client secret



For more information about the properties, see *StandardOauth2AccessTokenProvider configurations*.

### JsonRecordSetWriter

You use this controller service as the writer, when writing the data as JSON array or one JSON object per line in NiFi. You can choose another writer depending on your requirements.

In general, you may accept the default configurations.

### What to do next

After you have finished configuring your controller services:

- Configure your source processor
- Configure your target processor

### Related Information

[NiFi documentation](#)

### StandardOauth2AccessTokenProvider configurations

Learn about how to configure the properties for the StandardOauth2AccessTokenProvider controller service.

- Authorization Server URL

The authorization server URL is the concatenation of the base URL and the token request service URL ( /services/oauth2/token).

- Username

The username is the email address registered in the Salesforce account in use.

- Password

For the password, a security token must be requested. The password is made up of the Salesforce account password and the security token, concatenated together without a space. Perform the following steps to get the security token:

1. Log in to Salesforce.
2. Go to Profile Settings .
3. Go to Reset My Security Token tab in the left-navigation pane.
4. Click Reset Security Token.

The security token will be sent to you in an email.

- Client ID

In order to fill the Client ID, you need to create a new Connected App within Salesforce.

1. Log in to Salesforce.
2. Navigate to Setup.
3. On the left search panel find App Manager.
4. Click New Connected App.
5. Configure the properties, and click Save.

After it is done, the Consumer Key goes to the Client ID property.

- Client secret

The client secret can be found on the same page, as described in the Client ID section, under the name of Consumer Secret.

## Configuring your source processor

You use the QuerySalesforceObject processor to get data from your Salesforce table. To do this, launch the Configure Processor window, specify the necessary configurations, and start the process to verify that you can view the Salesforce table.

### Before you begin

- You must have built the dataflow.
- You must have created your controller services.

### Procedure

1. Launch the Configure Process window, by right-clicking the processor and selecting Configure.
2. Go to the Properties tab.

**Configure Processor** | QuerySalesforceObject 1.17.0-SNAPSHOT

Stopped

SETTINGS SCHEDULING **PROPERTIES** RELATIONSHIPS COMMENTS

Required field

Property	Value
URL	https://cloudera-2f-dev-ed.my.salesforce.com
API Version	54.0
sObject Name	Account
Field Names	Id,BillingAddress
Read Timeout	15 s
OAuth2 Access Token Provider	StandardOAuth2AccessTokenProvider →
Record Writer	JsonRecordSetWriter →
Create Zero Record FlowFiles	false
Age Field	No value set
Custom WHERE Condition	No value set

CANCEL APPLY

3. Enter the value for URL.

You can obtain the base URL from the Salesforce account profile menu and prefix it with https://.

4. Enter the value for API Version.

This is the Salesforce REST API version. For more information, see *Introduction to REST API*.

5. Specify the value for sObject.

The sObject name specifies the Salesforce object to be queried.

6. Specify the value(s) for Field Names.

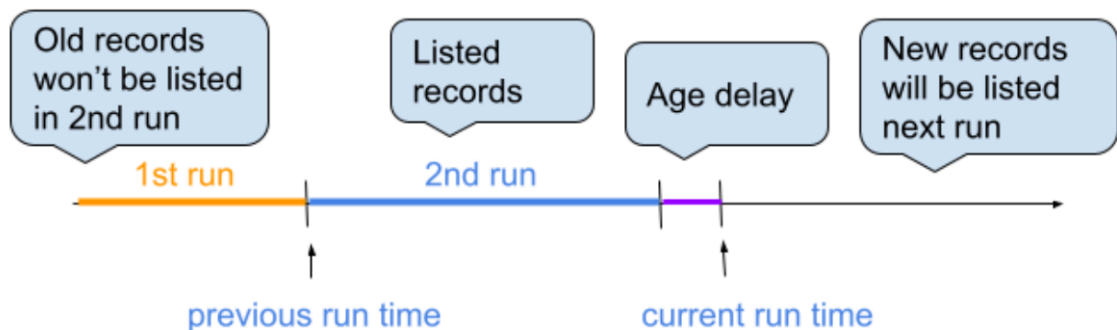
The field names are the Salesforce object's fields. Due to Salesforce limitations, it cannot be empty in order to obtain every field of an entity.

## 7. Set the value for Age Field.

The age property is important if you only want to query and process a record once. It provides a sliding window that starts with the processor's prior runtime and ends with the current runtime minus the age delay. Only records that are within the sliding window are queried and processed.

You need to define how to compare the records with the runtime. The Age Field property must be a datetime field of the queried object and this is subject to the condition that it is greater than the processor's previous runtime but less than the current runtime (for example, LastModifiedDate).

The first run, for example, queries records whose LastModifiedDate field is earlier than the current runtime. The second run looks for records with LastModifiedDate fields that are later than the previous runtime but earlier than the current runtime.



The Initial Age Filter is used to specify a specific datetime that sets the start of the sliding window from which processing starts in the first processor run. The format must meet the Salesforce SOQL standards. The Initial Age Filter is used to provide a specific datetime that sets the beginning of the sliding window from which processing occurs in the first processor run. Here, the format must adhere to the Salesforce SOQL requirements. For more information, see *SOQL and SOSL Reference*.

The purple line in the above image represents Age Delay. It moves the time of the records to be processed earlier than the current runtime if necessary.

### Results

You can confirm that you have configured QuerySalesforceObject correctly by starting the processor and confirming that you can view the Salesforce table.

### What to do next

After you have finished configuring your source processor, proceed to:

- Configure your target processor.
- Confirm your data flow is running.

### Related Information

[Introduction to REST API](#)

[SOQL and SOSL Reference](#)

## Configuring your target processor

In your dataflow that is pulling data from a Salesforce database, configure the required processor to handle data pooling from remote tables. To do this, launch the processor configuration window and provide the configurations appropriate for your use case.

### Procedure

1. Right-click the target processor and click Configure Processor.

2. Configure the properties of your target processor that suits your use case.

### What to do next

After you have configured your target processor, proceed to confirming your data flow success.

## Confirming your dataflow success

Learn how you can confirm success for your dataflow.

### About this task

You put the processor to the test after you enable the Controller Service. In JSON format, you obtain the ID and BillingAddress of all accessible Account data.

```
[
  {
    "Id": "0017Q0000003JfjcQAC",
    "BillingAddress": {
      "city": "San Francisco",
      "country": "US",
      "countryCode": null,
      "postalCode": "94087",
      "state": "CA",
      "stateCode": null,
      "street": "The Landmark @ One Market",
      "geocodeAccuracy": null
    }
  },
  {
    "Id": "0017Q0000003BwDdQAK",
    "BillingAddress": null
  },
  {
    "Id": "0017Q0000003ZdgiQAC",
    "BillingAddress": {
      "city": "Budapest",
      "country": "Hungary",
      "countryCode": null,
      "postalCode": "1232",
      "state": "Budapest",
      "stateCode": null,
      "street": "Apro u. 12/B",
      "geocodeAccuracy": null
    }
  }
]
```

When querying date, time, or datetime fields, Cloudera recommends setting the format properties on the RecordWriter service.