Use Case: Launch Prefilled Forms from a Salesforce Button

Download the PDF of this article.

In this Article

Related Articles

Overview

Salesforce users often need to collect additional information related to an existing Account without asking users to re-enter data that's already stored in Salesforce. By adding a prefilled form as a custom button on the Account record, users can launch a form with Account information already populated.

This approach is ideal for scenarios where users need to collect additional information to streamline data collection, reduce manual entry, and improve data accuracy.

Overview of the Process

Adding a prefilled FormAssembly form button to Salesforce involves two main steps: configuring your form to retrieve Salesforce data and creating a Salesforce button that launches the form with the appropriate record information.

Here's how it works:

- **Configure the form:** The FormAssembly connector is set up to identify a Salesforce record and determine which fields should be used to prefill the form.
- **Create the Salesforce button:** A custom button is added to a Salesforce record page and configured with a form URL that passes the current record's ID to FormAssembly.
- **Launch the prefilled form:** When a user clicks the button, FormAssembly uses the record ID to retrieve the Salesforce data and populates the mapped fields before displaying the form.

This creates a seamless process where users can start a form directly from Salesforce without manually entering information that already exists in their Salesforce records.

Build Your Prefill Connector

Before creating your Salesforce button, configure your form so it knows how to retrieve and populate data from an Account record.

Choose the instructions below based on the Salesforce integration you use.

Method 1- Using the Form-Native Salesforce Connector

When you are using the Form-Native Salesforce Connector, you can follow the steps below to retrieve an existing

Account record and use it to prefill your form.

Step 1: Configure a Lookup Query

The Lookup query tells FormAssembly which Account record to retrieve.

To configure the Lookup, click **Configure Your First Lookup** and select **Account** from the **Select an Object** menu. Then set the Lookup on the **Account ID** to equal the **Unsafe Query Parameter** *acctid.*

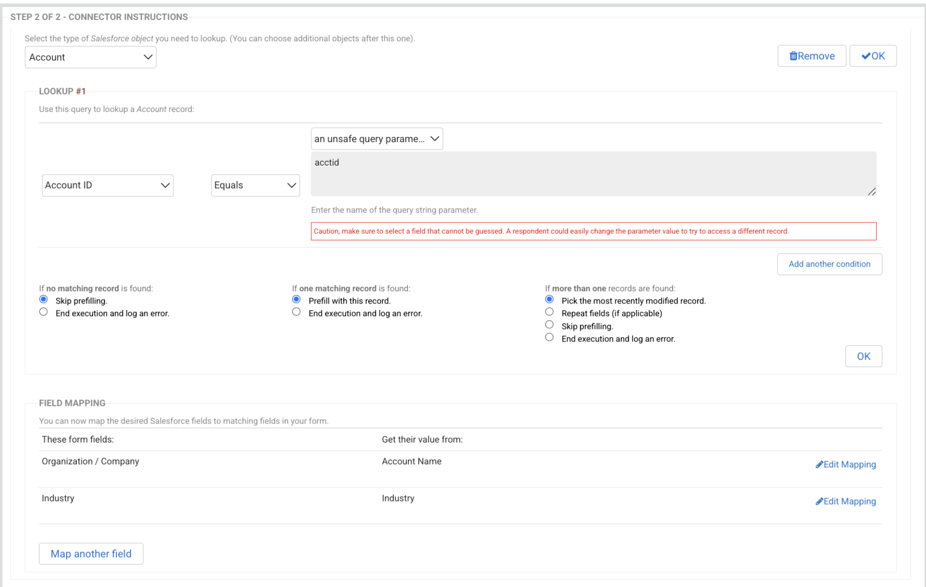
Match your selections with the example below:

- **If no matching record is found:** Skip this object and any dependent.
- **If one matching record is found:** Update record (default).
- **If more than one record is found:** Choose from the options below based on your needs.
    - Pick the most recently modified record — pulls the record that was last saved
    - Repeat fields (if applicable) — pulls in all matching records to be prefilled
    - Skip prefilling — nothing is prefilled into the form if multiple are found
    - End execution and log an error — the prefill connector will stop and show an error on the form.

Step 2: Map Salesforce Fields to Form Fields

After configuring your Lookup, select the form fields you want to be prefilled with their corresponding Salesforce fields.

Your form is now configured to populate fields using the Salesforce record returned by your Lookup.



Method 2 - Using the Salesforce Workflow Connector

When you are using the Salesforce Workflow Connector, you can follow the steps below to retrieve an existing Account record and use it to prefill your form.

Step 1: Create a Workflow URL Parameter

The workflow URL parameter allows FormAssembly to receive the Account ID from the Salesforce button and use that value to identify the Account record that should be retrieved.

To configure the URL parameter, open your Workflow Builder and navigate to **Workflow Properties**. Add a new URL

Parameter, provide a name such as *acctid,* and choose from the following options

- **Unsecure parameter:** The parameter is not signed in the Workflow URL.
- **Secure parameter:** The parameter is signed in the Workflow URL.

For more information on workflow parameters, refer to our documentation.

Step 2: Configure the Salesforce Workflow Connector

Add a Salesforce Workflow Connector step at the beginning of your workflow and set the action to **Lookup a Record**.

Select **Account** as the Salesforce object to retrieve. In the Lookup Query section, add a condition that matches the Account ID field to the workflow variable created in the previous step.

Match your Lookup settings with the below:

- **Lookup Field:** Account ID
- **Operator:** equals
- **Value:** Workflow variable *acctid*

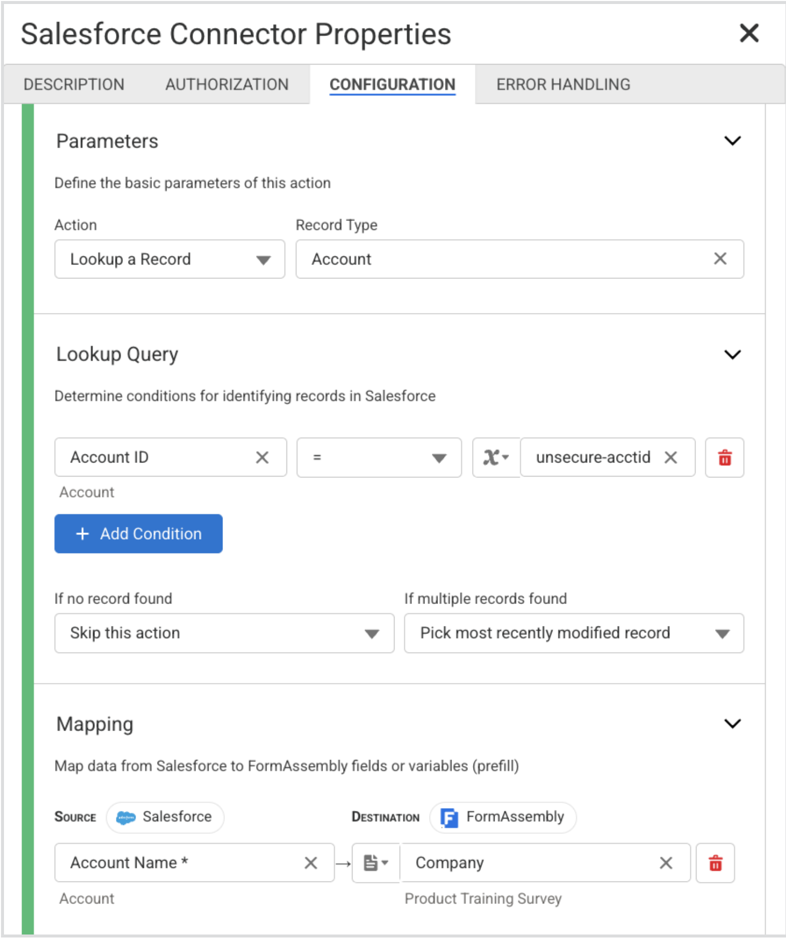
Then configure how the connector should react if no record or multiple records are found; match them with the following:

- **If no record is found:** Skip this action
- **If more than one record is found:** This depends on your needs.
    - Skip this action
    - End execution and return an error
    - Pick most recently modified record
    - Select all records

Step 3: Map Salesforce Fields to Form Fields

After configuring your Lookup, map the Salesforce Account fields you want to retrieve to their corresponding form fields.

Your workflow is now configured to retrieve the Account record passed from Salesforce and use it to populate fields in your form.



Add the Prefilled Form Button to the Account Record

After configuring your form, the next step is to create a custom Salesforce button that passes the Account ID of the record a user is currently on to your form.

Create a Custom Button

To create the button, open Salesforce Setup and navigate to **Object Manager > Account > Buttons, Links, and Actions**. Click **New Button or Link** and configure the button with the following settings:

- **Label:** Enter a name for your button, such as *Submit Feedback*.
- **Button Type:** Select a button type.
  - List button — Appears on a related list on an object record page.
  - Detail page link — Appears in the Links section of the record details on an object record page.
  - Detail page button — Appears in the action menu in the highlights panel of a record page.
- **Behavior:** Choose a behavior.
  - Display in new window — this will open a new tab in the browser with the link
  - Display in existing window with or without sidebar — this will open the link in the existing tab
- **Content Source:** Select URL.

Refer to the Salesforce documentation if you need additional guidance on creating buttons or links.

Configure the Prefilled Form URL

Start with the URL for your FormAssembly form:

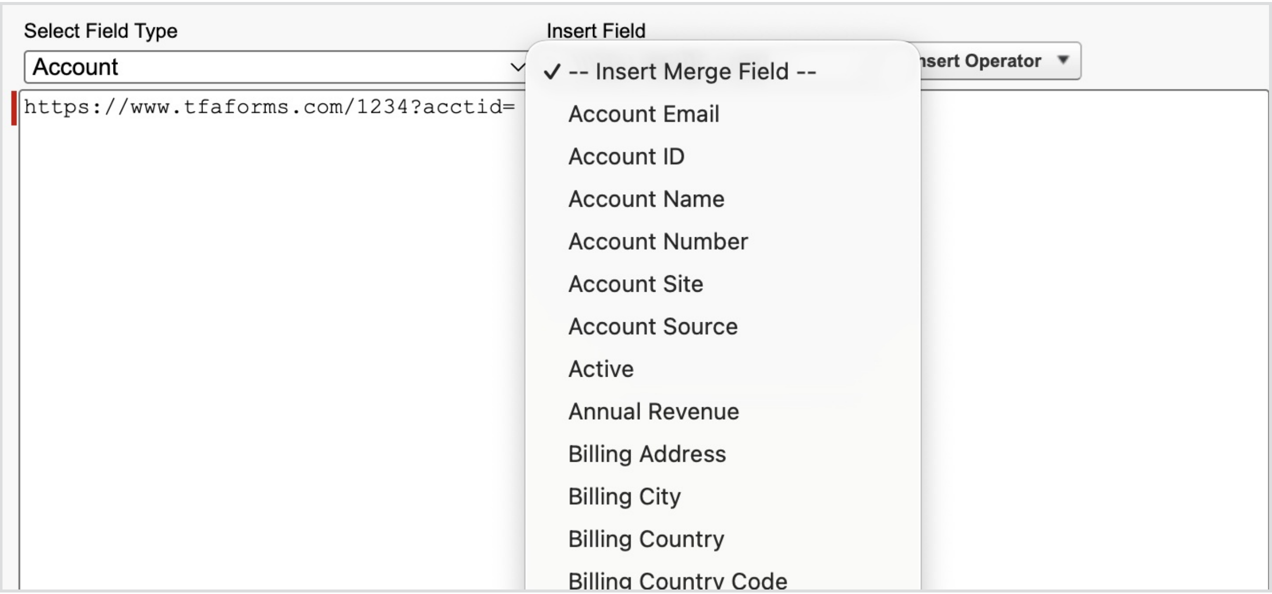
https://[INSTANCE]/[FORM_ID]

- Replace [INSTANCE] with your FormAssembly instance and [FORM_ID] with the ID of your form, or just copy and paste your form's URL.
- Example: https://www.tfaforms.com/1234

Add the query parameter configured in your prefill connector:

https://www.tfaforms.com/1234?acctid=

Next, add the Salesforce Account ID merge field to the end of the URL using the Insert Field option.



Your final URL should look similar to:

https://www.tfaforms.com/1234?acctid={!Account.Id}

When a user clicks the button from an Account record, Salesforce replaces **{!Account.Id}** with the current Account ID. FormAssembly then uses that value to retrieve the matching Account record and populates the mapped fields.

Add the Button to the Account Page Layout

After creating the button, you will need to add it to the Account page layout where you want users to access the form.

Refer to the Salesforce documentation if you need additional guidance on editing page layouts.

Result

Users can now click the custom button directly from an Account record to open a FormAssembly form with Account information already populated. The form retrieves the current Account using its Salesforce Account ID, reducing manual data entry, improving data accuracy, and allowing users to complete and submit the form quickly.