Lightning Workshop: Get started with Flow Builder

Abstract - We know Flow can be intimidating. So we've partnered with Jennifer Lee, automation all-star and Lightning Champion, to help you understand what Flow is all about and to help you get started with Flow Builder. In this workshop, you will learn about the foundational building blocks of Flow and build your very first flow.

Before we get started, let's cover the basics!

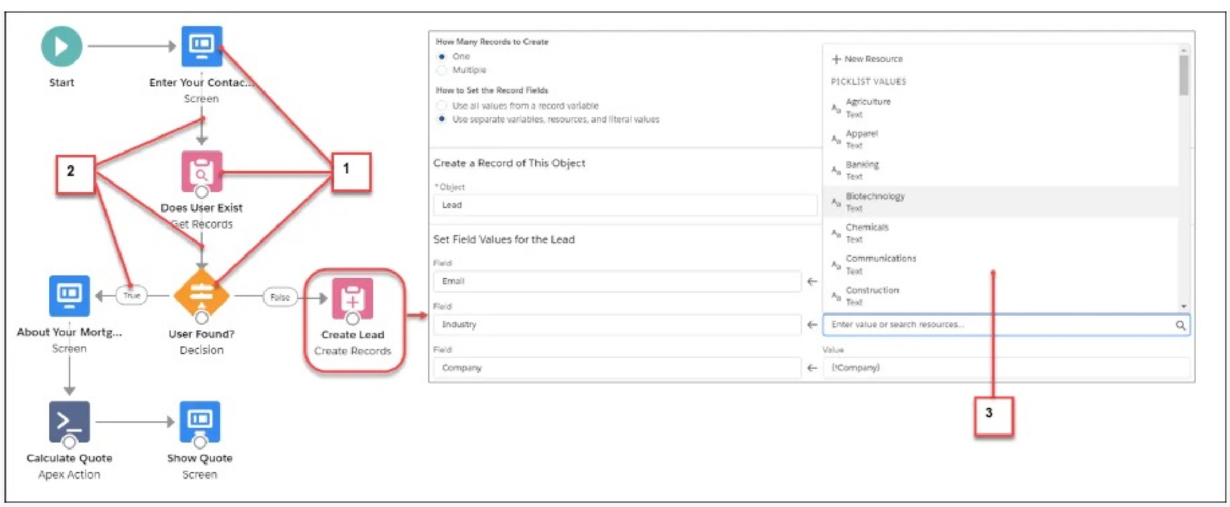
Step 1: Review Flow

Flow is an application inside of Salesforce that automates a business process by collecting data and performing operations in your org or an external system.

(demo driver: Navigate to Setup>Flows>New Flow. Select Screen Flow >Create to walk through these pieces.)

Every flow is made up of three building blocks:

- Elements (1) appear on the canvas. To add an element to the canvas, click it or drag it there from the toolbox.
- Connectors (2) define the path that the flow takes at runtime. They tell the flow which element to execute next.
- Resources (3) are containers that represent a given value, such as field values or formulas. You can reference resources throughout your flow. For example, look up an account's ID, store that ID in a variable, and later reference that ID to update the account.

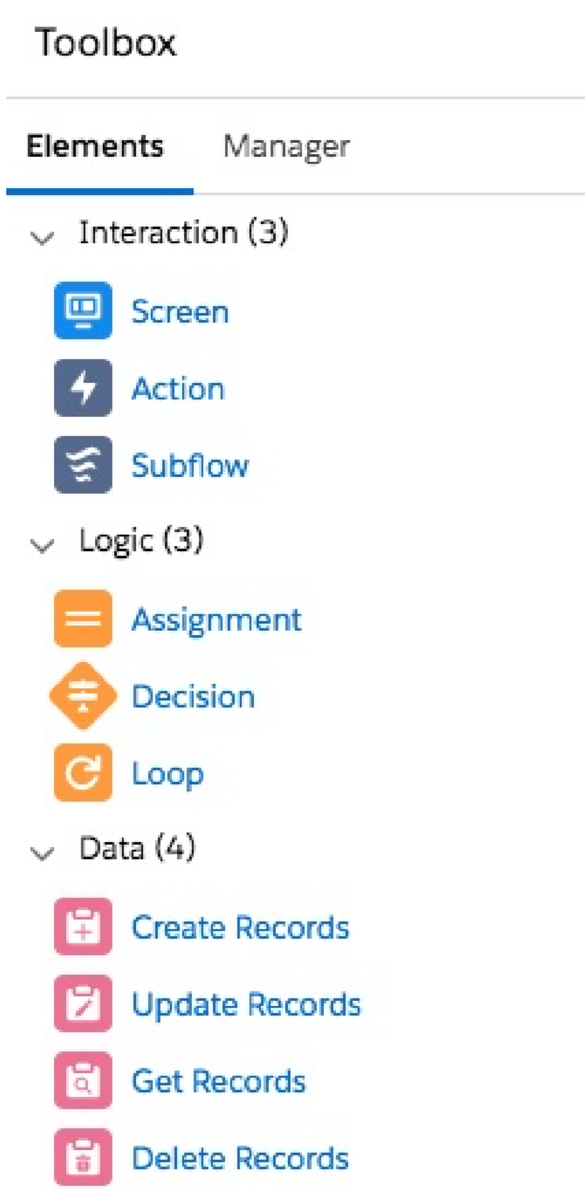


The **toolbox** contains the elements and resources you'll use to build your flow.

- From the Elements tab, add new elements (like the ones below) to your flow:
  - ***Screen*:** Displays a screen to the user running the flow, to display or collect information from the user.
  - ***Action*:** Allows you to Post to Chatter, Send Email, Submit For Approval, Activate or

Deactivate a Permission Set, Global or object-specific actions provided by Salesforce

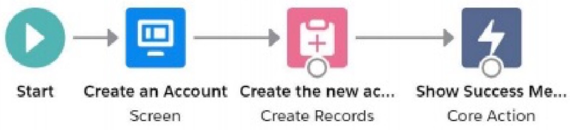
- ***Subflow***: Calls another flow in your org. Use this element to reference modular flows and simplify the overall architecture of your flow.
- ***Assignment***: Sets or changes values in variables, collection variables, sObject variables, and sObject collection variables.
- ***Decision***: Evaluates conditions and routes users accordingly based on the decision outcomes. Equivalent to an if-then statement.
- ***Loop***: Iterates through a collection one item at a time, and executes actions on each item's field values.
- ***Create Records***: Creates one record by using individual field values that you specify or creates records using field values from a multiple collection record or record variable.
- ***Update Records***: Finds all records meeting the specified criteria and updates them with individual field values you specify OR updates records using the field values from a multiple collection record variable, or one record using the field values from a record variable.
- ***Get Records***: Finds the first record meeting the specified criteria and assigns the record's field values to individual flow variables or individual fields on record variables OR finds records to assign their field values to a multiple collection record variable, or finds a record to assign its field values to an record variable.
- ***Delete Records***: Deletes a single record meeting the specified criteria OR deletes in a multiple collection record variable.



From the Manager tab, create resources, such as variables, stages, and choices, to use within your flow. Or view a list of all elements and resources that you've added to the flow.

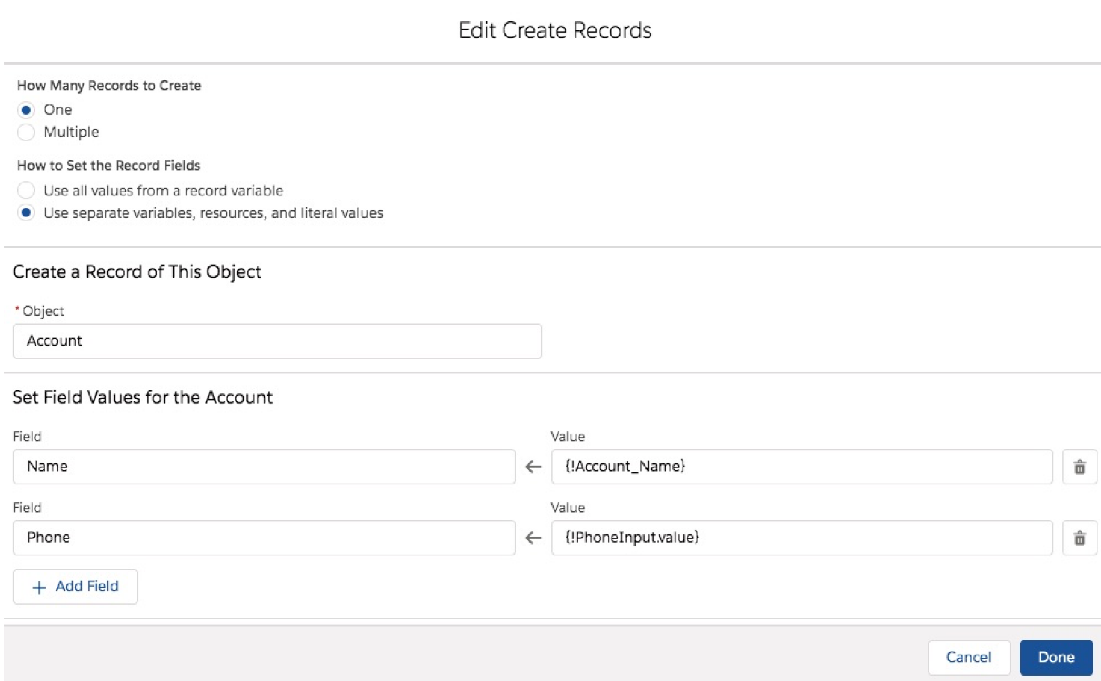
Step 2: Build a Flow

We have been tasked to create a flow that gathers account name and account phone number from the user. The system will then create a new account record with the information, and we will show a message to the user letting the user know he/she has successfully created a record.

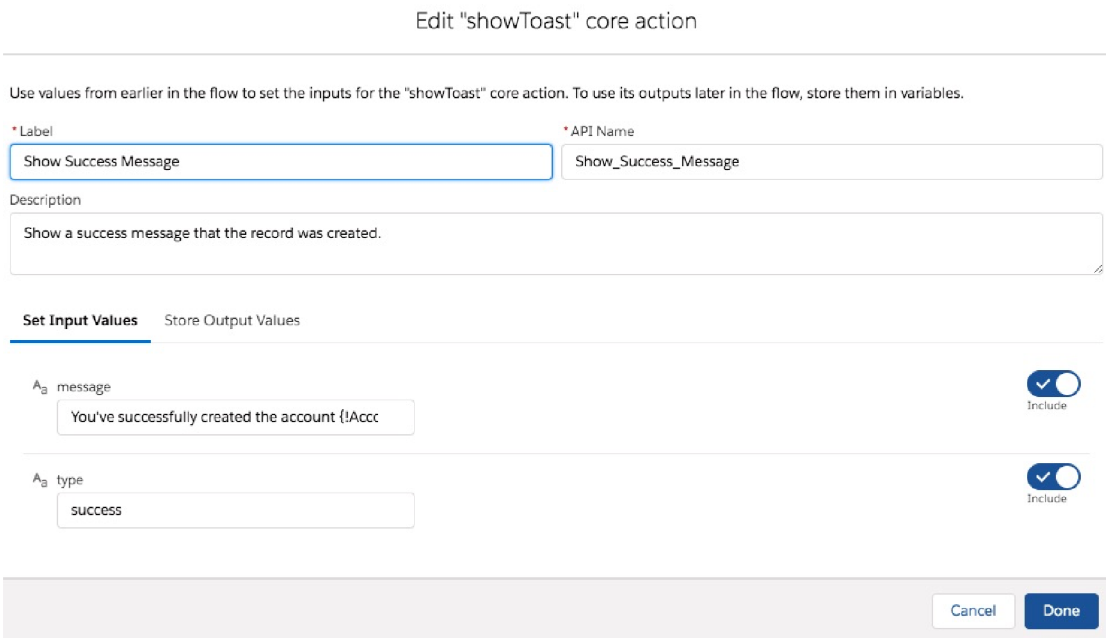


- Create a new Flow.
  - Navigate to **Setup**
  - Search for **Flow** in the Quick Find Search.
  - Click on **Flows**
  - Select **New Flow**
  - Select **Screen Flow**
  - Click **Create**
- From our review, we learned a screen element displays a screen to the user running the flow, to display or collect information from the user. **Drag a Screen element** onto the Flow to gather the account name and account phone number for the new account
  - Set Label: **Create an Account**
  - Set API Name: **Create_an_Account**
  - Provide a Description: **Create a new account**
  - Expand Control Navigation so users can only move forward:
    - **Deselect Previous and Pause**
  - Locate the **Text component** and drag it onto the screen palette.
    - Set Label: **Account Name**
    - Set API Name: **Account_Name**
    - Set Required: **Checked**
  - Locate the **Phone component** and drag it onto the screen palette.
    - Set API Name: **PhoneInput**
    - Set Label: **Phone**
    - Set Placeholder Text: **Enter 555-555-5555**
    - Set Required: **{!$GlobalConstant.False}**
  - **Done**
- Add a **Create Records** element to create the account record with the account name and account phone number provided by the user
  - Set Label: **Create the new account**
  - Set API Name: **Create_the_new_account**
  - Provide a Description: **Auto create the account based on the values from the input screen**
  - Set How Many Records to Create: **One**
  - Set How to Set the Record Fields: **Use separate variables, resources, and literal values**
  - Set Object: **Account**
  - Set Field: **Name**
  - Set Value: **{!Account_Name}** (Found under Screen Components)

- Select **+ Add Field**
- Set Field: **Phone**
- Set Value: **{!PhoneInput.value}**(Found under Screen Components, select PhoneInput, then select value)
- Select **Done**



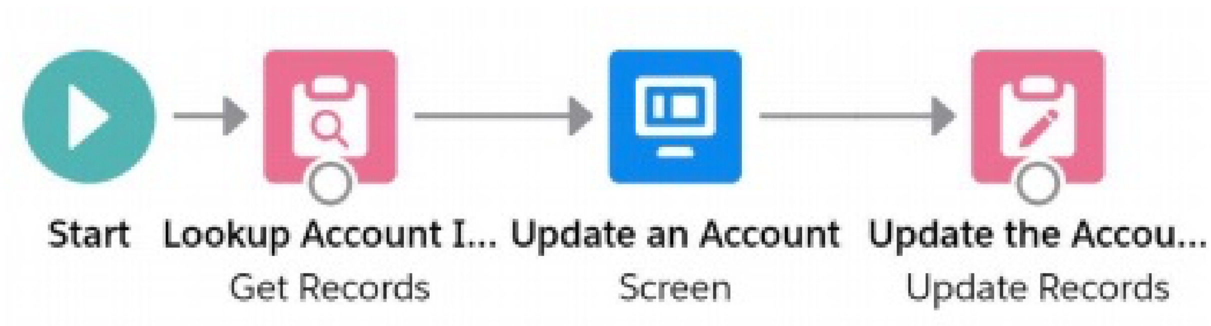
- Add an **Action** element that will call the showToast local action to show a success message that the record has been created.
  - In the Action window, type and select **showToast**.
  - Set Label: **Show Success Message**
  - Set API Name: **Show_Success_Message**
  - Provide a Description: **Show a success message that the record was created.**
  - Set Message toggle: **Include**
  - Set Message: **You've successfully created the account {!Account_Name}.**
  - Set Type toggle: **Include**
  - Set Type: **success**
  - Select **Done**



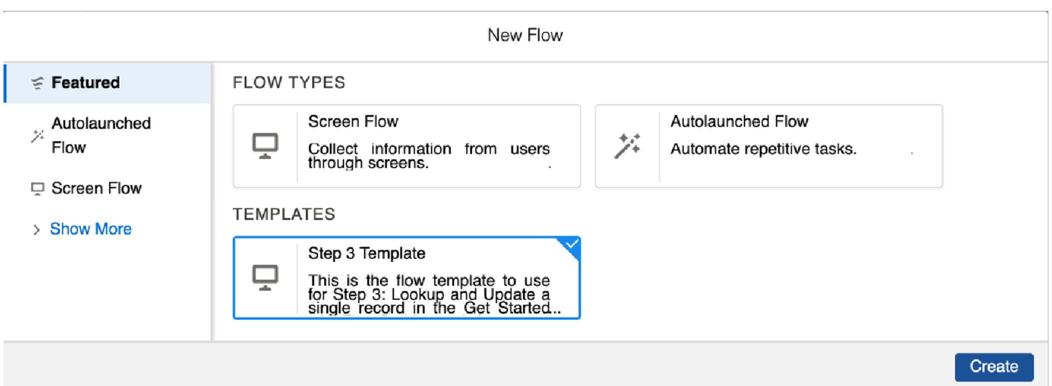
- Draw a connector from Start to the Screen element
- Draw a connector from the Screen element to the Create Records element
- Draw a connector from the Create Records element to the showToast local action element
- Select **Save**
  - Set Flow Label: **Create a Record**
  - Set Flow API Name: **Create_a_Record**
  - Provide a Description: **Create a new single record**
  - Set the Type: **Screen Flow**
- Click **Activate**
- Navigate to **App Manager**, edit the **Sales** app (Developer Name: LightningSales)
- Add a Flow component to the Utility Bar. Set the **Create a Record** flow in the properties of the Flow component.
  - In the App, select **Utility Items**
  - In Utility Items, **Add Utility Item**
  - Select **Flow**
  - Change the Label: **Create Account**
  - Scroll down and set Flow: **Create a Record**
- Select **Save**
- Back
- Navigate to the Sales app and open the Flow component in the Utility Bar

Step 3: Lookup and Update a single record

After creating a record, we have been asked to create a flow to pre-populate the account name and account phone number then update the account name and/or account phone number.

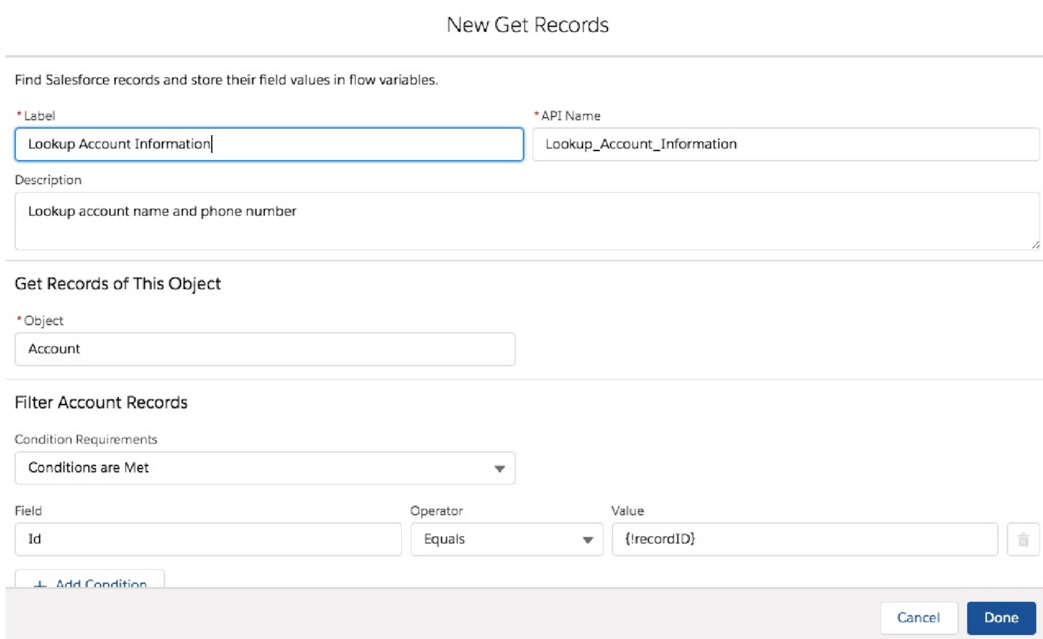


- Create a new flow.
  - Navigate to **Setup**
  - Search for **Flow** in the Quick Find Search.
  - Click on **Flows**
  - Select **New Flow**
  - Select **Step 3 Template** under Templates
    - A flow template is a base business process that you can clone to create customized business processes. Unlike non-template flows, if you install a template from a managed package, you can open and clone it (which is a great way to get started with Flow). Even if you build your own templates and don't use managed packages, marking a flow as a template clearly identifies it as a base business process. For example, suppose your company needs a flow that differs slightly for each country where you do business. You can create or install a template for the base business process and then clone it to create each country-specific business process.



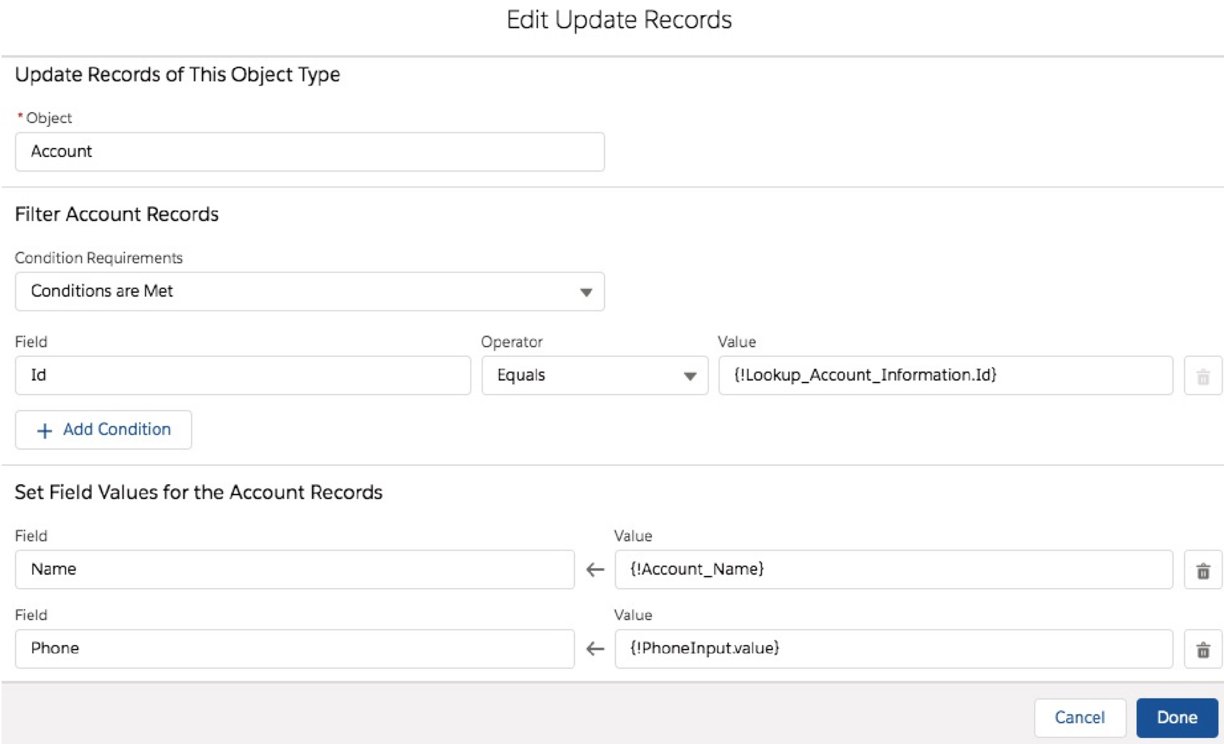
  - Click **Create**
- Add a **Get Records** element to lookup the account name and account phone number for the account record. (Note: We will pass the account record ID from the Lightning record page later in the exercise.)
  - Set Label: **Lookup Account Information**
  - Set API Name: **Lookup_Account_Information**
  - Provide a Description: **Lookup account name and phone number**

    - Set Object: **Account**
    - Set Condition Requirements: **Conditions are Met**
    - Set Field: **Id**
    - Set Operator: **Equals**
    - Set Value: **+ New Resource**
    - Let's create the variable.
        - Set Resource Type: **Variable**
        - Set API Name: **recordId**
        - Provide a Description: **Stores the recordId from the Lightning record page**
        - Set Data Type: **Text**
        - Available for input: **Checked**
        - Select **Done**



    - Set How Many Records to Store: **Only the first record**
    - In the Field box, select **Phone**
    - Click **Add Field** and select **Name**
    - Select **Done**
- Open the **Screen element** that shows the account name and account phone number and prompts the user for update.
    - Update the Label: **Update an Account**
    - Update the API Name: **Update_an_Account**
    - Update the Description: **Update an account**
    - Set the Account Name Text component to the following.
        - Set Label: **Account Name**
        - Set API Name: **Account_Name**

        - Set Default Value: **{!Lookup_Account_Information.Name}**(Under Record (Single) Variables, select Account from Lookup_Account_Information, then select Name)
    - Set the Phone component to the following.
        - Set Label: **Phone**
        - Set API Name: **Phone**
        - Set Value: **Select {!Lookup_Account_Information.Phone}** (Under Record (Single) Variables, select Account from Lookup_Account_Information, then select Phone)
    - Select **Done**
- Select the **Create Records element** and click on the trash can. Repeat this for the Action element. These steps are not needed.
- Add an **Update Records element** to update the existing account with the user inputted account name and phone number.
    - Set Label: **Update the Account Info**
    - Set API Name: **Update_the_Account_Info**
    - Provide a Description: **Update the account info from the screen input**
    - Set How to Find Records to Update and Set Their Values: **Specify conditions to identify records, and set fields individually**
    - Set Object: **Account**
    - Set Condition Requirements: **Conditions are Met**
    - Set Field: **Id**
    - Set Operator: **Equals**
    - Set Value: **{!Lookup_Account_Information.Id}** (Under Record (Single) Variables, select Account from Lookup_Account_Information, then select Id)
    - Set Field: **Name**
    - Set Value: **{!Account_Name}** (Found under Screen Components)
    - Select **+ Add Field**
    - Set Field: **Phone**
    - Set Value: **{!PhoneInput.value}**(Found under Screen Components, select PhoneInput, then select value)
        - If you do not see this option, check your PhoneInput screen component on the screen element. As you scroll down, look to see if the box to “Manually assign variables (advanced)” is checked. If it is, uncheck it and select **Delete.** In Winter ‘20, Salesforce automatically creates variables in a screen component or a Get Records element to store the output values. You no longer have to create and assign variables, but you can still opt to do so.
    - Select **Done**



- Remove the connector between Start and the Screen element.
- Draw a connector from Start to the Get Records element.
- Draw a connector from the Get Records element to the Screen element
- Draw a connector from the Screen element to the Update Records element.
- Select **Save**
  - Set Flow Label: **Update a Single Record**
  - Set Flow API Name: **Update_a_Single_Record**
  - Provide a Description: **Update information on a single record**
  - Set the Type: **Screen Flow**
- Click **Activate**
- Navigate to the **Sales** app
- Open an Account record
- Click on the gear icon, select **Edit Page** to open Lightning App Builder.
- Drag the **Flow component** to the right-hand section
  - Set Flow: **Update a Single Record**
  - Pass record ID into this variable: **Checked**
  - Select **Save**
  - Activate as **Org Default**
  - Select **Back**

Congratulations! You have learned about Flow and have started to work in the new Flow Builder. This is only just the beginning, though. To continue building on your Flow expertise, complete the Build Flows with Flow Builder trail on Trailhead!