

Icon Picker documentation

This listing includes LWC component Icon Picker which allows to select Icon from SLDS Icons Library

It can be used in the following use cases

1. Component can be placed in Lightning App Builder "App page" and used in Dynamic Interactions
2. Component can be placed in Screen Flows
3. Component can be reused in other LWC

Using IconPicker in App page

Create a new App page or place the component on an existing App Page.

Go to App Builder in Setup



Setup
 Home
 Object Manager

User Interface
 Lightning App Builder

Didn't find what you're looking for?
Try using Global Search.

Lightning App Builder

The Lightning App Builder provides an easy to use graphical interface for creating custom Lightning pages for Salesforce Lightning Experience and mobile app. Lightning pages are built using Lightning components—compact, configurable, and reusable elements that you can drag and drop into regions of the page in the Lightning App Builder.

View: All [Create New View](#)

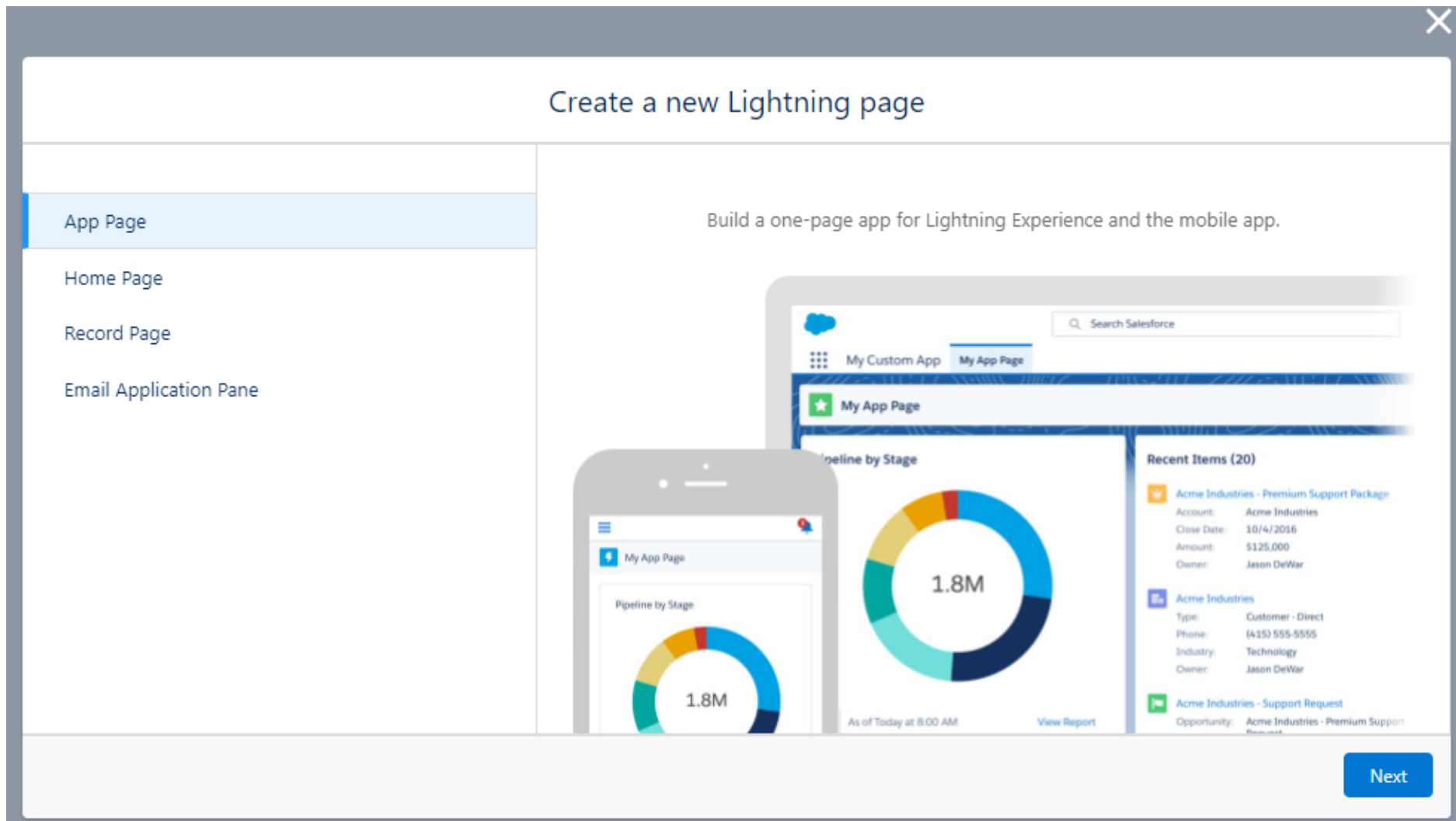
A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other | **All**

Lightning Pages

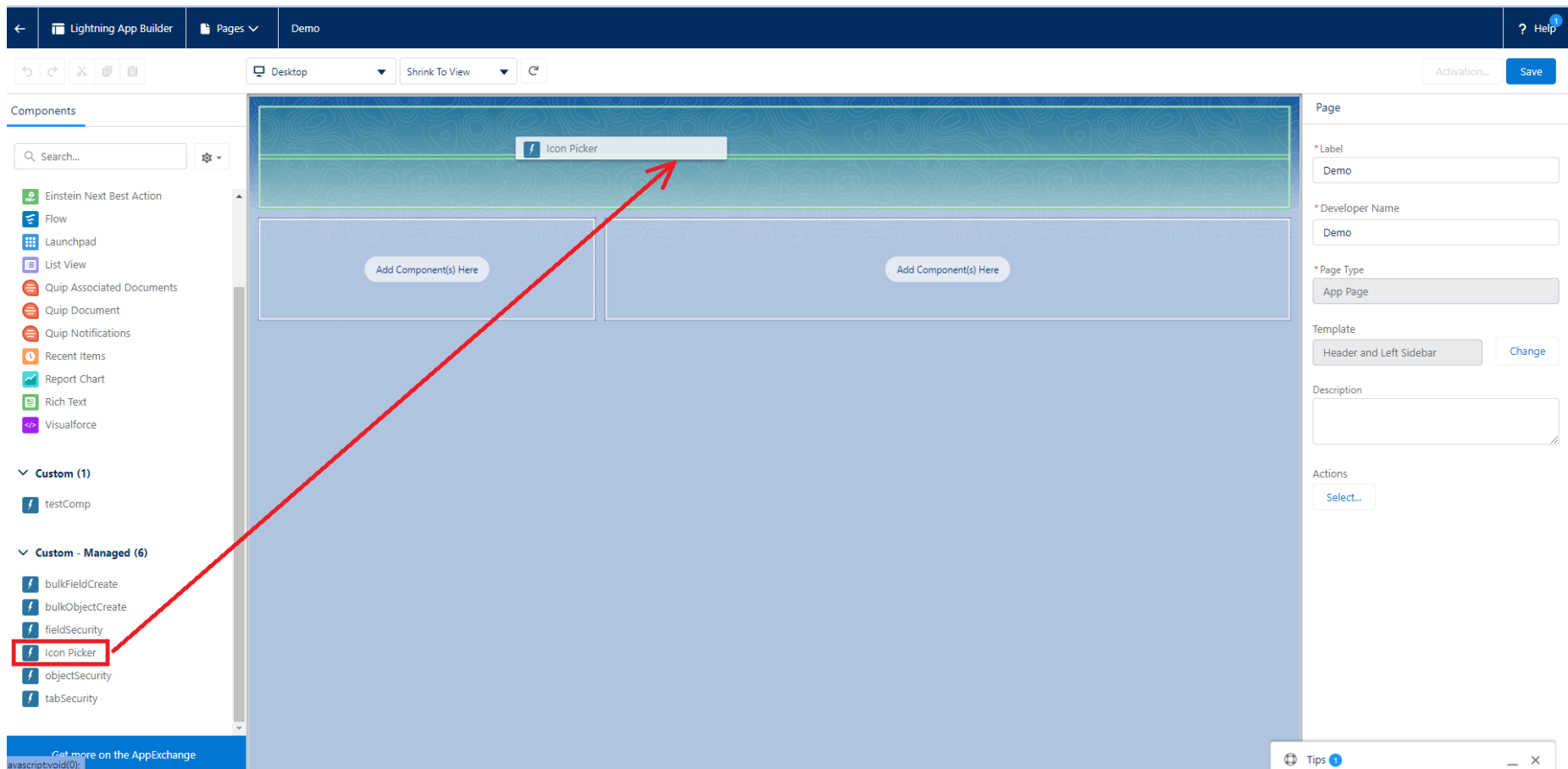
New

Action	Label ↑	Name	Namespace Prefix	Description	Type	Created By	Last Modified By
Clone View	Bulk Field Creation Manager	BulkFieldCreationManager	bofperman		App Page	BDovh 11/1/2021, 9:01 AM	BDovh 11/1/2021, 9:01 AM
Clone View	Bulk Field Permission Manager	BulkFieldPermissionManager	bofperman		App Page	BDovh 11/1/2021, 9:01 AM	BDovh 11/1/2021, 9:01 AM
Clone View	Bulk Object Creation Manager	BulkObjectCreationManager	bofperman		App Page	BDovh 11/1/2021, 9:01 AM	BDovh 11/1/2021, 9:01 AM
Clone View	Bulk Object Permission Manager	BulkObjectPermissionManager	bofperman		App Page	BDovh 11/1/2021, 9:01 AM	BDovh 11/1/2021, 9:01 AM
Clone View	Bulk Tab Permission Manager	BulkTabPermissionManager	bofperman		App Page	BDovh 11/1/2021, 9:01 AM	BDovh 11/1/2021, 9:01 AM
Edit Clone Del	Test	Test			App Page	BDovh 1/1/2022, 5:53 AM	BDovh 1/1/2022, 5:56 AM

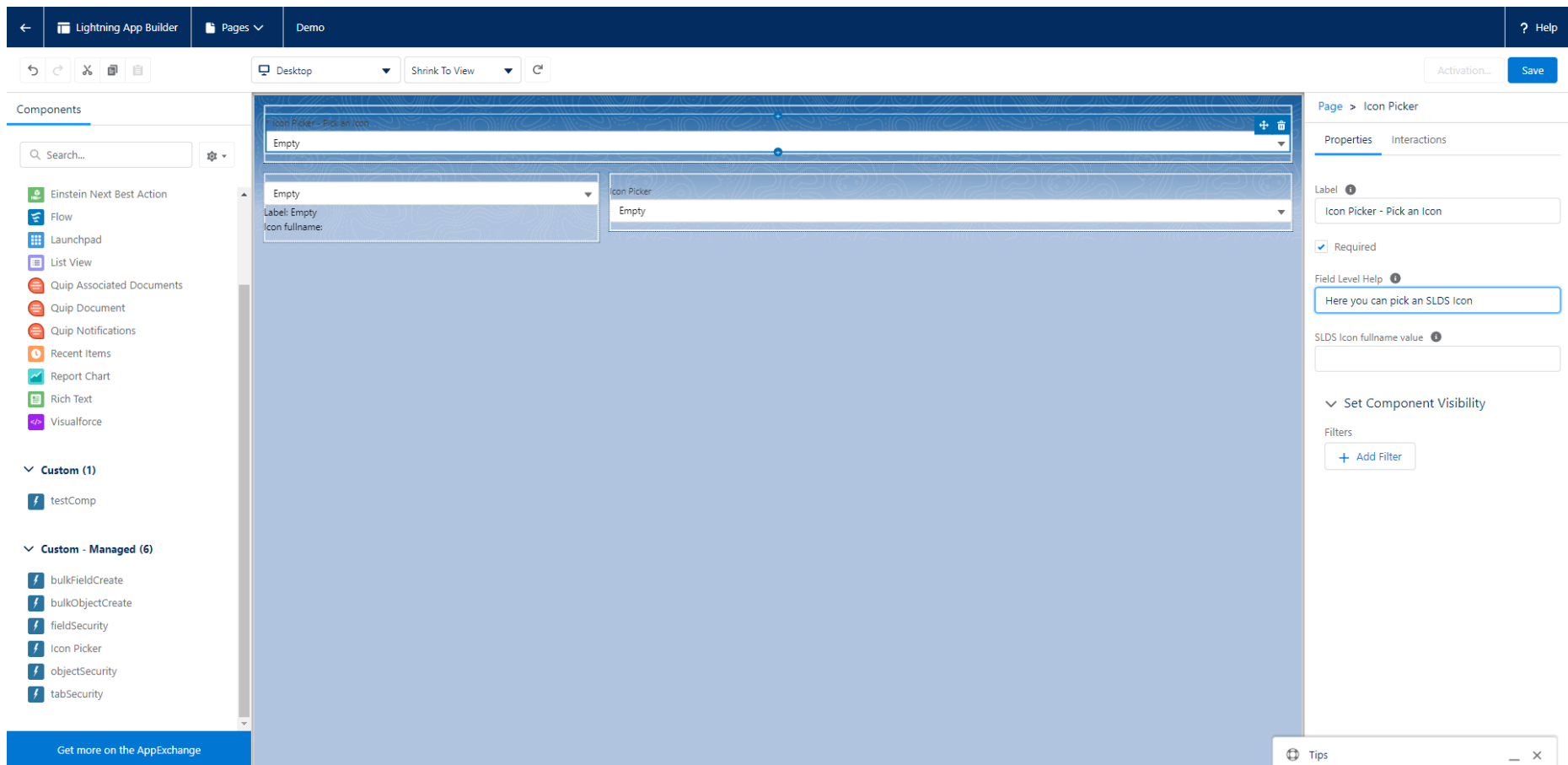
Click a New button to create a new App Builder page or use Edit to edit existing App Page
If you create a new Page, select App Page and click Next button



Define a label for the new page, for example, Demo, and select any layout, for example, header and left sidebar. Find Icon Picker component in the “Custom - Managed” section and Drag and Drop that component to any place, like header or sidebar.

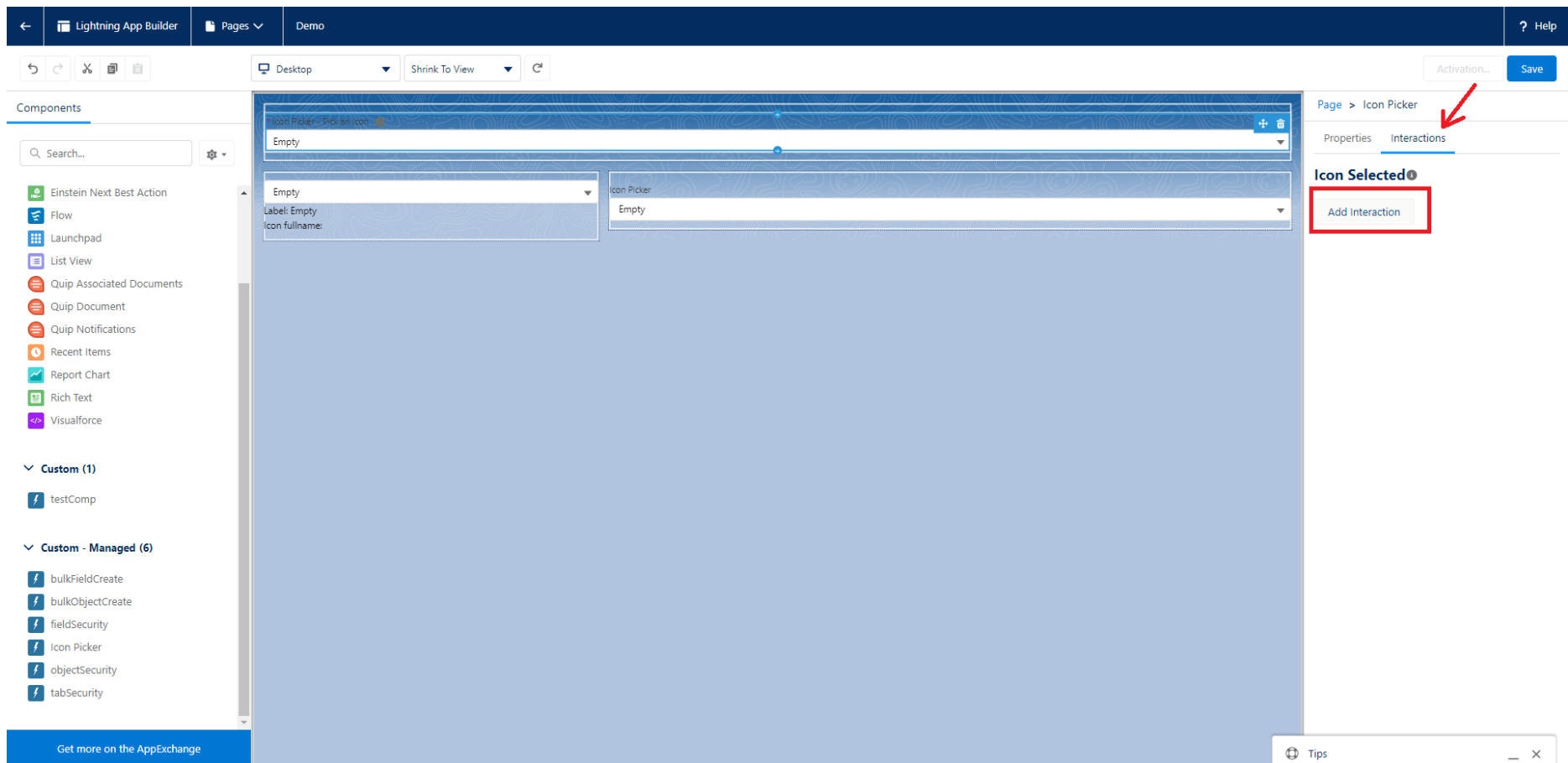


You can add other components, your custom unpackaged components or the same component in another section. You can read about creating custom unpackaged LWC components embedding Icon Picker in the section "[Use IconPicker in Custom LWC Development](#)". In the Properties pane, you can set header label and field level help and mandatory indicator, for example, use `Icon Picker - Pick an Icon` like header and `Here you can pick an SLDS Icon` as field level help. If you check the required checkbox, the red asterisk will appear on the screen.

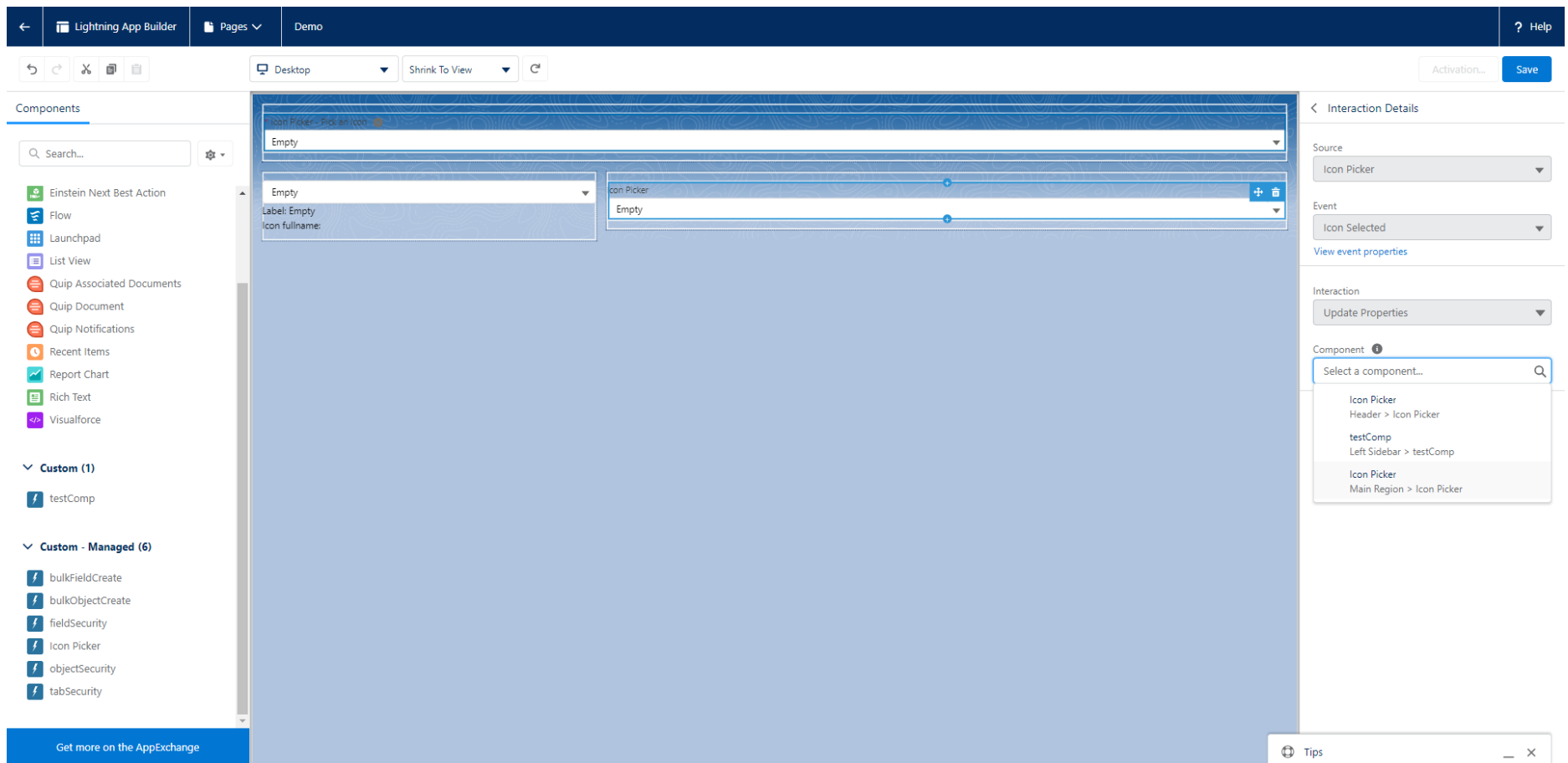


On the Interactions pane, it is possible to configure the dynamic interactions.

For example, let's set source the IconPicker component in the header section and target the custom TestComp component in the left sidebar and the IconPicker component in the main section. To start, click Interaction pane and then click Add Interaction button.



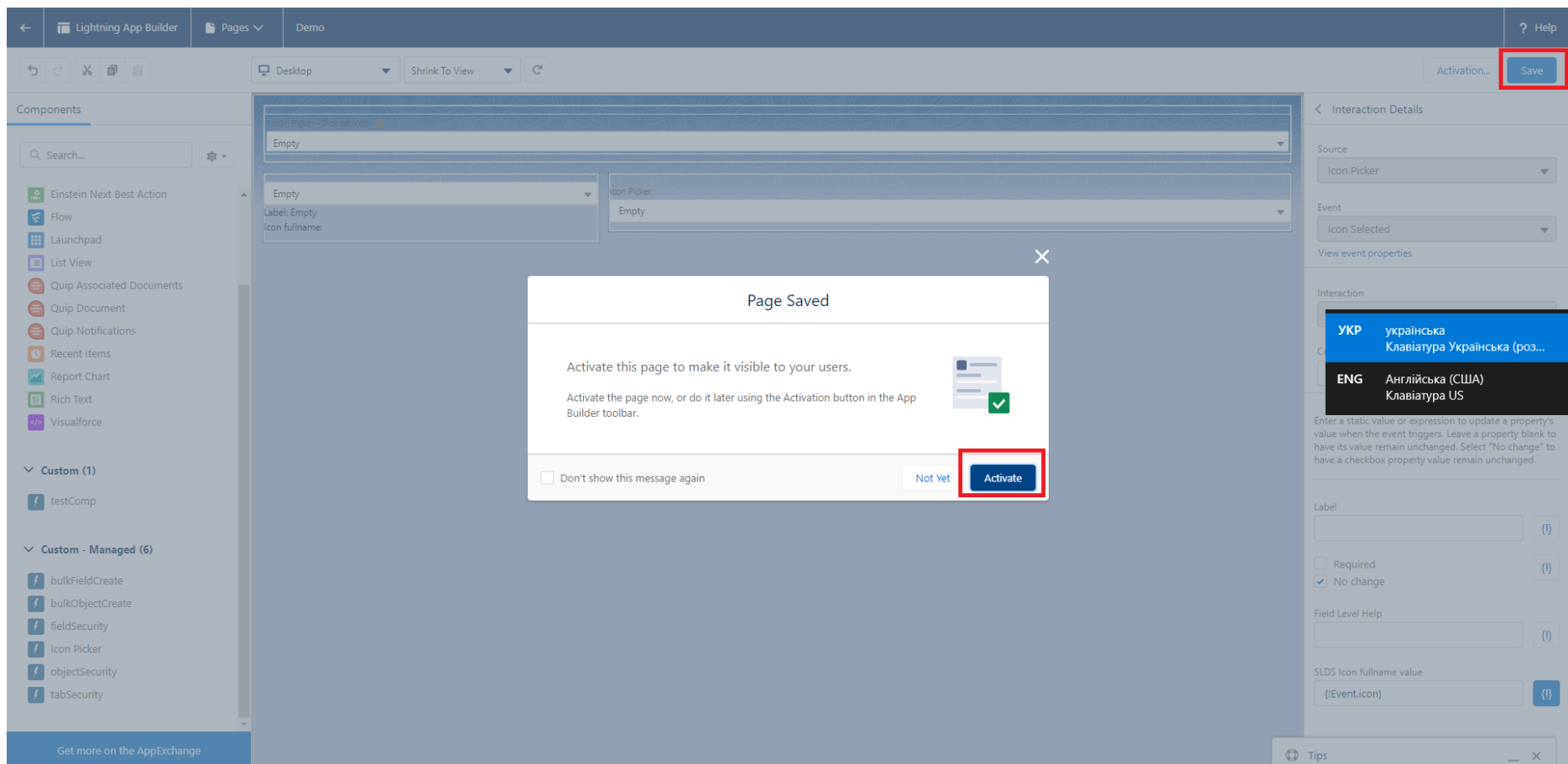
While the source component is selected in the App Builder, in the component selector select target component IconPicker component from Main Region.



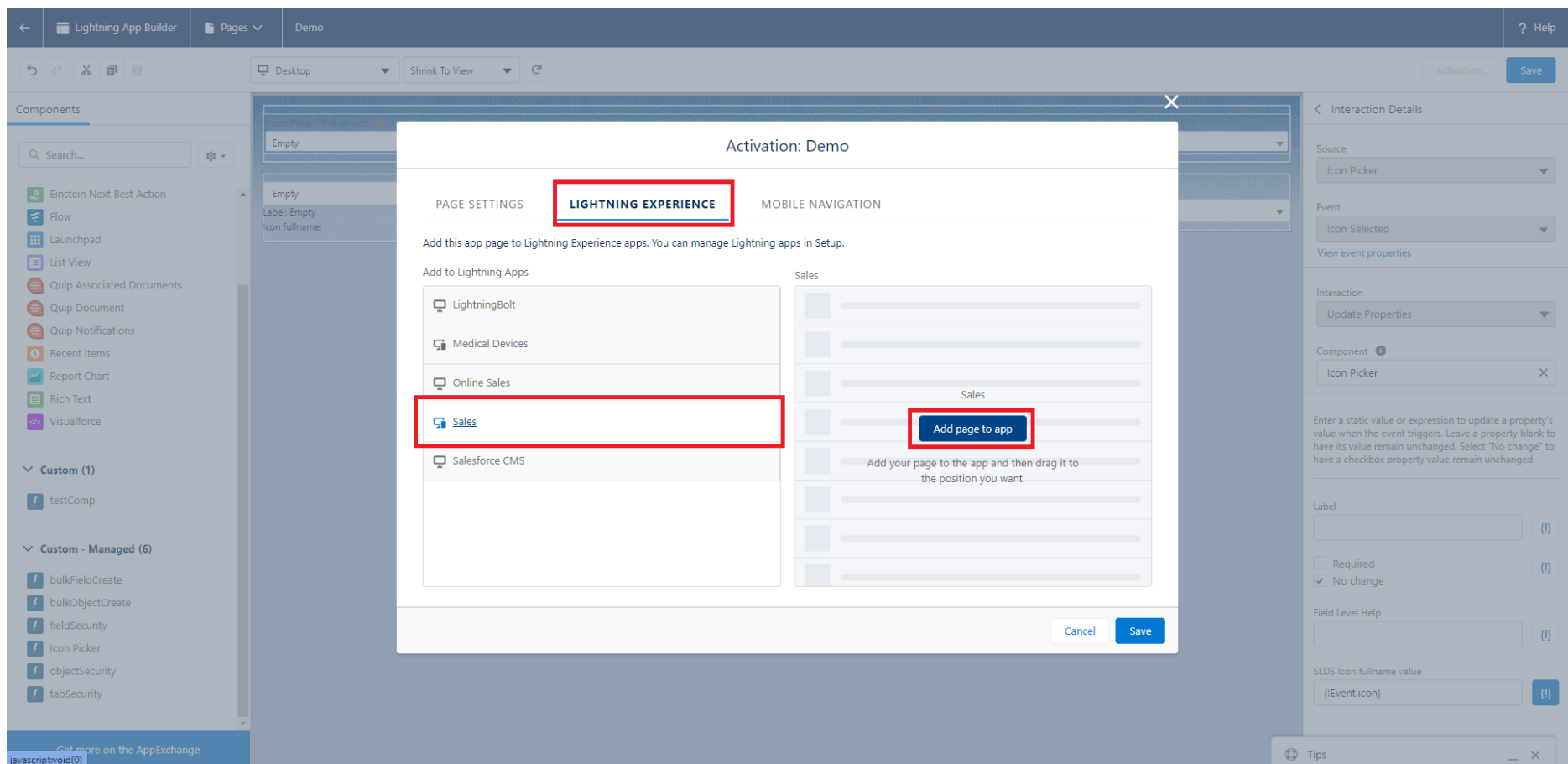
Then set `{!Event.icon}` in SLDS Icon fullname value attribute

The screenshot displays the Lightning App Builder interface. The top navigation bar includes 'Lightning App Builder', 'Pages', 'Demo', and a 'Help' icon. Below the navigation bar, there are icons for undo, redo, delete, and a 'Desktop' view selector. The main workspace shows a canvas with a blue background and a patterned border. On the left, the 'Components' panel lists various components: Einstein Next Best Action, Flow, Launchpad, List View, Quip Associated Documents, Quip Document, Quip Notifications, Recent Items, Report Chart, Rich Text, Visualforce, Custom (1) with testComp, and Custom - Managed (6) with bulkFieldCreate, bulkObjectCreate, fieldSecurity, Icon Picker, objectSecurity, and tabSecurity. The 'Icon Picker' component is selected and placed on the canvas. The right sidebar shows the 'Interaction Details' for the 'Icon Picker' component. It includes fields for 'Source' (Icon Picker), 'Event' (Icon Selected), and 'Interaction' (Update Properties). The 'Component' field is set to 'Icon Picker'. Below these fields, there is a text area for 'Enter a static value or expression to update a property's value when the event triggers. Leave a property blank to have its value remain unchanged. Select "No change" to have a checkbox property value remain unchanged.' The 'Label' field is empty, and the 'Field Level Help' field is empty. The 'SLDS Icon fullname value' field is set to '{Event.icon}'. At the bottom right, there is a 'Tips' button.

and then click the Save button. If you created a new Page, Activate the page and set access to profiles and add it to application



Select if you want to add the page to all profiles or only System Administrator and add to some app like Sales and click Add page to app



and then click Save button

Lightning App BuilderPagesDemo

DesktopShrink To View

Components

Search...

Einstein Next Best Action

Flow

Launchpad

List View

Quip Associated Documents

Quip Document

Quip Notifications

Recent Items

Report Chart

Rich Text

Visualforce

Custom (1)

testComp

Custom - Managed (6)

bulkFieldCreate

bulkObjectCreate

fieldSecurity

Icon Picker

objectSecurity

tabSecurity

Empty

Label: Empty

Icon fullname:

Activation: Demo

PAGE SETTINGSLIGHTNING EXPERIENCEMOBILE NAVIGATION

Add this app page to Lightning Experience apps. You can manage Lightning apps in Setup.

Add to Lightning Apps

LightningBolt

Medical Devices

Online Sales

Sales

Salesforce CMS

Sales

Remove page

Home

Demo

Test

Opportunities

Leads

Tasks

Files

Accounts

Cancel

Save

Interaction Details

Source

Icon Picker

Event

Icon Selected

View event properties

Interaction

Update Properties

Component

Icon Picker

Enter a static value or expression to update a property's value when the event triggers. Leave a property blank to have its value remain unchanged. Select "No change" to have a checkbox property value remain unchanged.

Label

Required

No change

Field Level Help

SLDS Icon fullname value

{Event.icon}

Get more on the AppExchange

Tips

Then click Back button

The screenshot shows the Lightning App Builder interface. At the top, there is a navigation bar with a back arrow, 'Lightning App Builder', 'Pages', 'Demo', and a 'Help' button. Below the navigation bar, there is a toolbar with icons for undo, redo, delete, and a 'Shrink To View' button. The main workspace is divided into three sections: a left sidebar for components, a central canvas, and a right sidebar for interaction details.

The left sidebar, titled 'Components', contains a search bar and a list of components. Under the 'Custom (1)' section, there is a component named 'testComp'. Under the 'Custom - Managed (6)' section, there are components: 'bulkFieldCreate', 'bulkObjectCreate', 'fieldSecurity', 'Icon Picker', 'objectSecurity', and 'tabSecurity'.

The central canvas shows a preview of the 'Icon Picker' component. It has a label 'Icon Picker - Pick an icon' and a dropdown menu with 'Empty' selected. Below the dropdown, there are two input fields: 'Label: Empty' and 'Icon fullname: Empty'.

The right sidebar, titled 'Interaction Details', contains settings for the selected component. It includes a 'Source' dropdown set to 'Icon Picker', an 'Event' dropdown set to 'Icon Selected', and an 'Interaction' dropdown set to 'Update Properties'. Below these, there is a 'Component' dropdown set to 'Icon Picker'. A text area for the interaction logic contains the text: 'Enter a static value or expression to update a property's value when the event triggers. Leave a property blank to have its value remain unchanged. Select "No change" to have a checkbox property value remain unchanged.' Below this, there are fields for 'Label' (empty), 'Required' (unchecked), 'No change' (checked), 'Field Level Help' (empty), and 'SLDS Icon fullname value' (set to '{!Event.icon}').

A red arrow points to the back arrow in the top right corner of the interaction details panel. A red box highlights the back arrow.

And then click again Add Interaction to add another interaction to testComp component

Lightning App Builder Pages Demo ? Help

Desktop Shrink To View

Activation... Save

Components

Search...

- Einstein Next Best Action
- Flow
- Launchpad
- List View
- Quip Associated Documents
- Quip Document
- Quip Notifications
- Recent Items
- Report Chart
- Rich Text
- Visualforce

Custom (1)

- testComp

Custom - Managed (6)

- bulkFieldCreate
- bulkObjectCreate
- fieldSecurity
- Icon Picker
- objectSecurity
- tabSecurity

Get more on the AppExchange

Page > Icon Picker

Properties Interactions

Icon Selected

Update Properties: Icon Picker

Add Interaction

Tips

Select TestComp or your custom component as target in Component selector

Lightning App Builder Pages Demo ? Help

Desktop Shrink To View

Activation... Save

Components

Search...

- Einstein Next Best Action
- Flow
- Launchpad
- List View
- Quip Associated Documents
- Quip Document
- Quip Notifications
- Recent Items
- Report Chart
- Rich Text
- Visualforce

Custom (1)

- testComp

Custom - Managed (6)

- bulkFieldCreate
- bulkObjectCreate
- fieldSecurity
- Icon Picker
- objectSecurity
- tabSecurity

Get more on the AppExchange

Interaction Details

Source: Icon Picker

Event: Icon Selected [View event properties](#)

Interaction: Update Properties

Component **?**

Select a component...

- Icon Picker
- Header > Icon Picker
- testComp
- Left Sidebar > testComp
- Icon Picker
- Main Region > Icon Picker

Tips

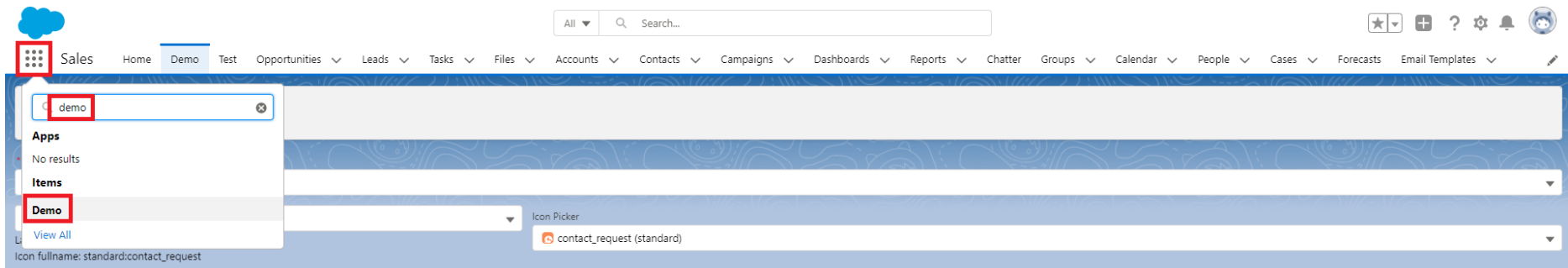
Set {!Event.icon} into destination attribute SLDS Icon fullname value

The screenshot displays the Lightning App Builder interface. At the top, a dark blue header bar contains navigation icons, the text "Lightning App Builder", "Pages", "Demo", and a "Help" icon. Below the header, a toolbar includes icons for undo, redo, delete, and a "Shrink To View" button. The main workspace is divided into three sections:

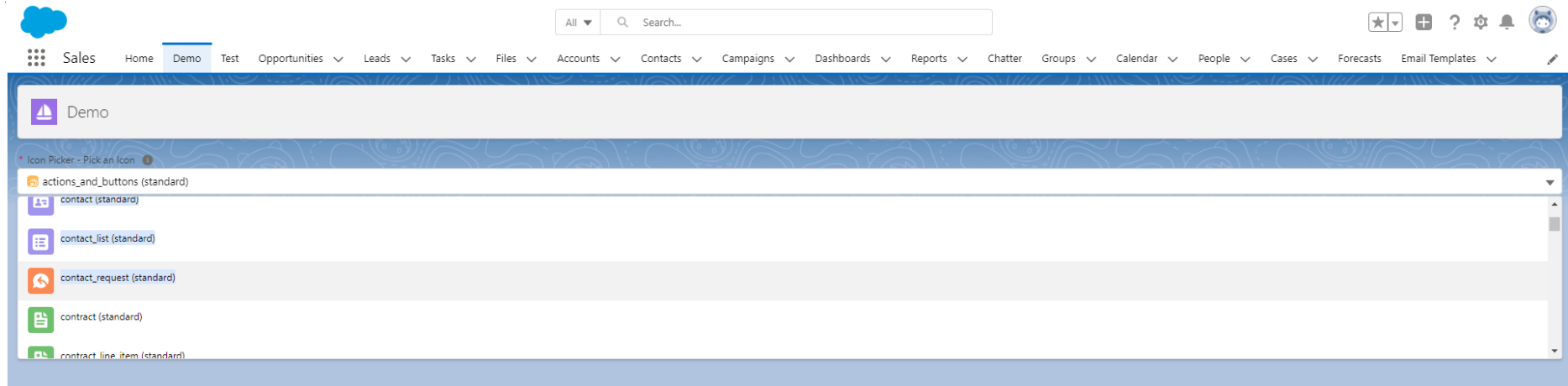
- Components:** A sidebar on the left with a search bar and a list of components. The list includes "Einstein Next Best Action", "Flow", "Launchpad", "List View", "Quip Associated Documents", "Quip Document", "Quip Notifications", "Recent Items", "Report Chart", "Rich Text", "Visualforce", and a "Custom (1)" section containing "testComp". Below this is a "Custom - Managed (6)" section with components like "bulkFieldCreate", "bulkObjectCreate", "fieldSecurity", "Icon Picker", "objectSecurity", and "tabSecurity".
- Canvas:** The central area showing a component being edited. It features a header bar with "Icon Picker - Pick an icon" and a dropdown menu. Below this is a section with "Label: Empty" and "Icon fullname:". To the right, there is an "Icon Picker" component with a dropdown menu.
- Interaction Details:** A panel on the right side. It includes a "Source" dropdown set to "Icon Picker", an "Event" dropdown set to "Icon Selected", and a link to "View event properties". Below this is an "Interaction" dropdown set to "Update Properties", a "Component" dropdown set to "testComp", and a text area for entering a static value or expression to update a property's value. At the bottom, there is a field for "SLDS Icon fullname value" with the expression "{!Event.icon}" and a "Tips" button.

At the bottom left, a blue button reads "Get more on the AppExchange". At the bottom right, a "Tips" panel is visible.

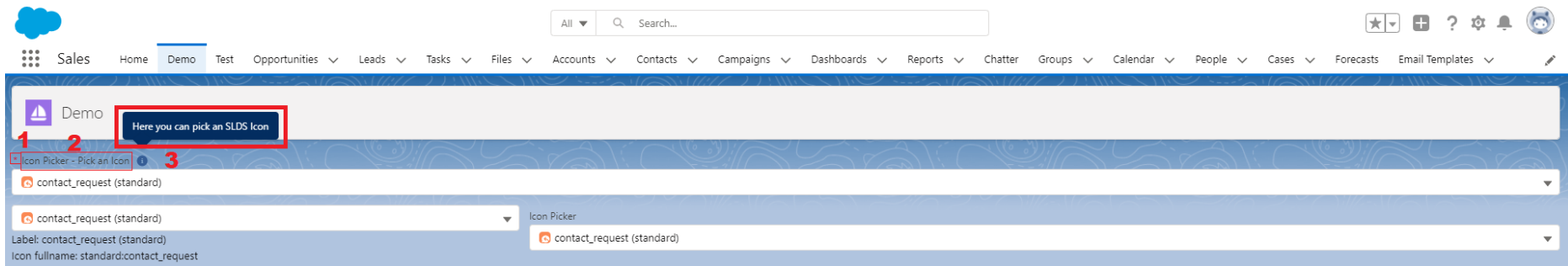
and then click the Save button again. Now let's see in action. Click on nine dots in the top left corner and input the App page name in search box and click on Demo in the results list.



On the demo tab, pick the icon and see that the picked icon is sent into two other components through Dynamic Interactions, so the same icon is now selected in two bottom components



Notice that configured header is displayed above the icon picker, the red asterisk is displayed since Required attribute checkbox is checked and the help text is displayed when mouse is hover the information icon



Using IconPicker in Screen Flows

It is possible to add IconPicker to Screen Flows.

Before doing that, please enable **Enable Lightning runtime for flows** in Setup/Process Automation Settings

Cloud logo

Search Setup

Setup Home Object Manager

process

Feature Settings

- Marketing
 - Lead Processes
- Sales
 - Sales Processes
- Service
 - Support Processes
- Process Automation
 - Approval Processes
 - Automation Home (Beta)
 - Flows
 - Next Best Action
 - Paused And Failed Flow Interviews
 - Post Templates
 - Process Automation Settings
 - Process Builder
- Workflow Actions
 - Email Alerts
 - Field Updates
 - Outbound Messages
 - Send Actions
 - Tasks
- Workflow Rules

Process Automation Settings

The default workflow user is required for scheduled paths in record-triggered flows and time-dependent actions in workflow rules. If the user who triggered the automation is no longer active, the default workflow user is associated with the actions that are executed by the automation. The default workflow user always runs schedule-triggered flows.

Default Workflow User

By default, the Automated Process User has no email address, so event processes, autolaunched flows with triggers, and orchestrations can't send emails for actions, errors, or notifications. To enable emails from the Automated Process User, specify an organization-wide email address for the system administrator profile.

Automated Process User Email Address

By default, the email approval sender is the approval submitter. To override the default sender, select an organization-wide email address.

Email Approval Sender **Approval Submitter**

Enabling email approval response lets users reply to email approval requests by typing APPROVE or REJECT in the first line and adding comments in the second line.

Enable email approval response ☐

By enabling the email approval response feature, you agree to allow Salesforce to process email approval responses, update approval requests for all active users in your organization, and update the approval object on behalf of your organization's users.

Let users pause flows when they need to wait for more information. Once you enable this setting, the Pause button appears on every screen that has Pause enabled.

Let users pause flows ☐

Let users resume interviews that are shared with them, either directly or via the role hierarchy. When disabled, interviews can be resumed only by the user who paused the interview or a flow admin who has view access.

Let users resume shared flow interviews ☒

Flows launched from a URL or from Setup use the Lightning runtime experience instead of the classic runtime experience. Only Lightning runtime supports two-column flow screens, custom screen components, and local actions.

Enable Lightning runtime for flows ☒

Change the required permission to view all charts on the Automation Home page in Setup. By default, users with the View Setup and Configuration permission can view all charts. When this setting is enabled, users with View Setup and Configuration can view only the Total Started Automations by Process Type chart.

Require the Manage Flow permission to view all Automation Home charts ☐

This setting affects Lightning Experience only. On the Flows page in Setup, use an enhanced flow list view that includes standard flows. On a separate page, view paused flows and process scheduled actions. When disabled or using Salesforce Classic, the Flows page displays custom flows, paused flows, and process scheduled actions. The separate page isn't available.

In Lightning Experience, use the enhanced Flows page and separate Paused and Scheduled Automations page ☒

This setting affects flows in API version 51.0 or earlier. When creating or updating records, the flow can use record variables to define the field values. If the running user doesn't have edit access to those fields, by default the flow fails to create or update the records. This setting filters the inaccessible fields from the request for Create Records and Update Records elements.

Filter inaccessible fields from flow requests ☐

Allows Apex code to set and remove approval process locks

Enable record locking and unlocking in Apex ☐

When this setting is enabled, users with the Manage Flow and View All Data permissions can specify the running user each time they debug a flow.

Let admins debug flows as other users ☐

When you debug a flow as another user, the flow's record changes and actions are performed as that user. Also, the user's profile and permission sets determine the object permissions and field-level access of the flow. However, flows that always run in system context ignore the user's object permissions and field-level access.

When a process or flow interview fails, Salesforce sends an email with details about everything that was executed and what failed. The details include the data that's involved in the process or flow, including user-entered data.

Find Flows in Setup and click the New Flow button to create a new flow or reuse the existing Screen Flow and click on it.



SetupHomeObject Manager

Search Setup

★+?⚙️🔔🐙

flows

Process Automation

Flows

Identity

Login Flows

Didn't find what you're looking for?
Try using Global Search.

SETUP
Flows

New Flow

Flow Definitions

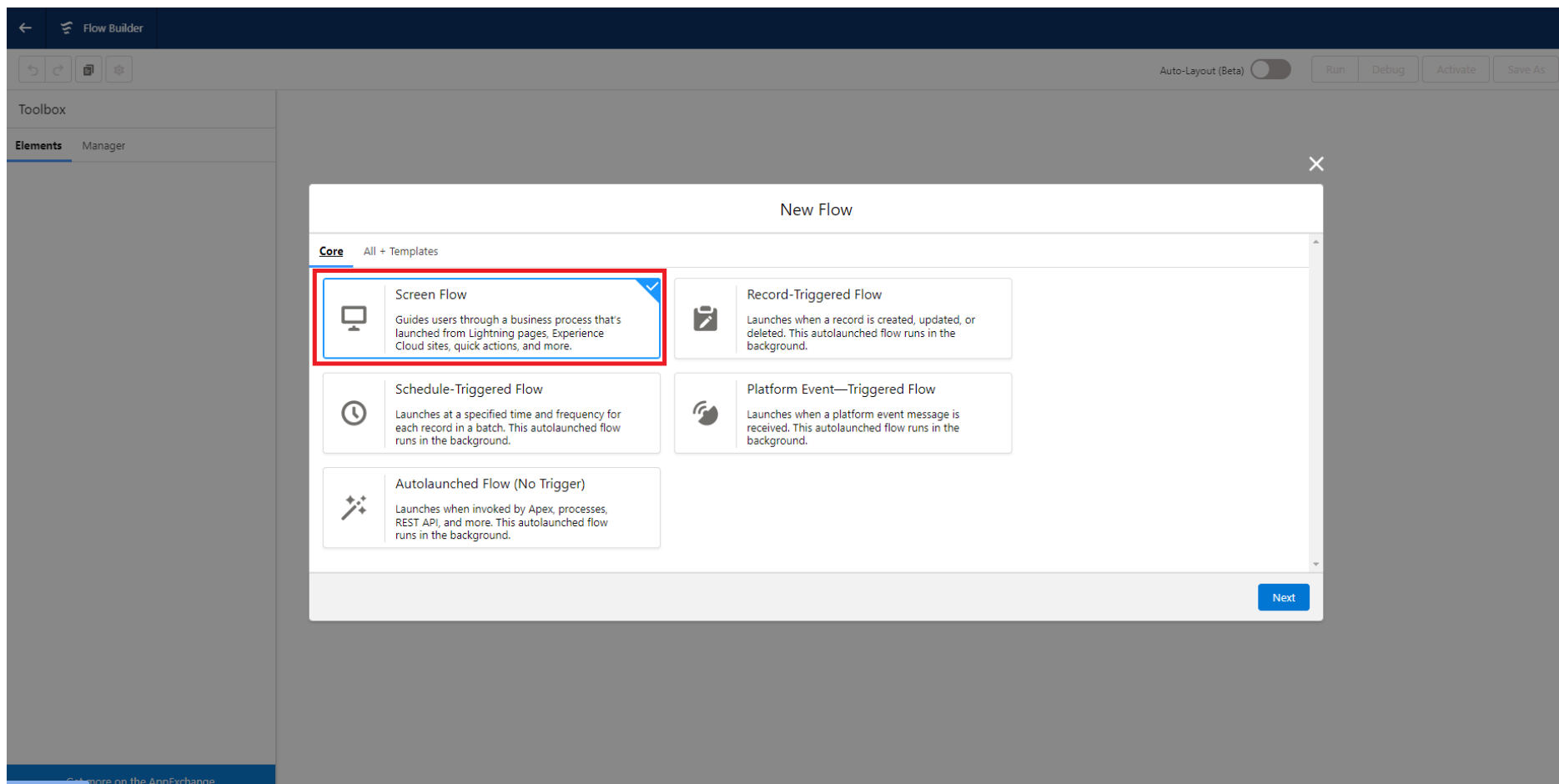
All Flows

5 items • Sorted by Flow Label • Filtered by All flow definitions • Updated a few seconds ago

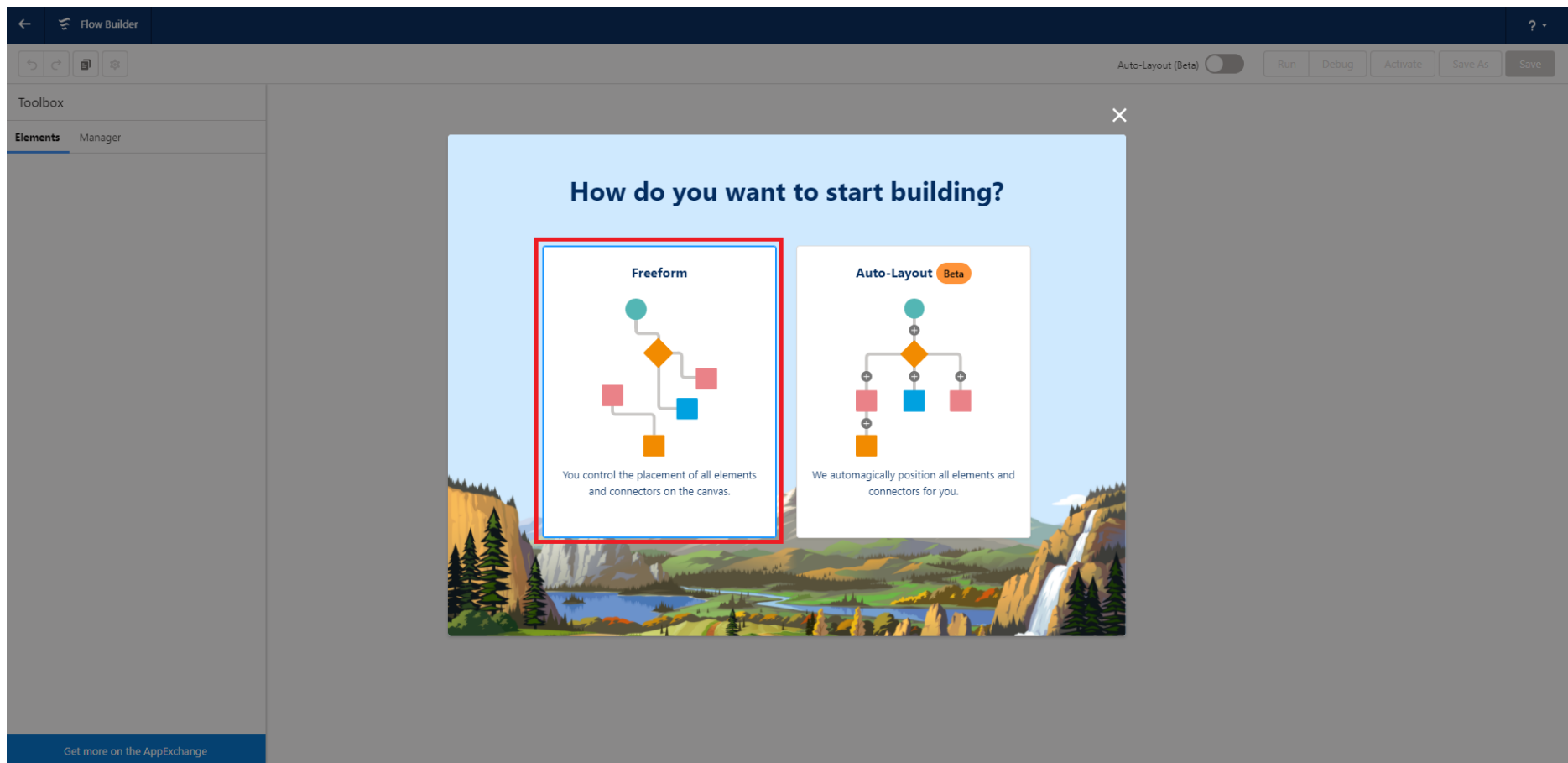
⚙️↺🔼

Flow Label ↑	Process Type	Active	Template	Package State	Package Name	Last Modified By	Last Modified Date
Create a Case	Screen Flow	✓	✓	Managed-Installed			
Reset Password	Screen Flow	✓	✓	Managed-Installed			
test1	Screen Flow	✓	□	Unmanaged		Bohdan Dovhan at Sales	1/1/2022, 6:08 AM
test2	Screen Flow	✓	□	Unmanaged		Bohdan Dovhan at Sales	1/1/2022, 6:11 AM
Verify Identity	Screen Flow	✓	✓	Managed-Installed			

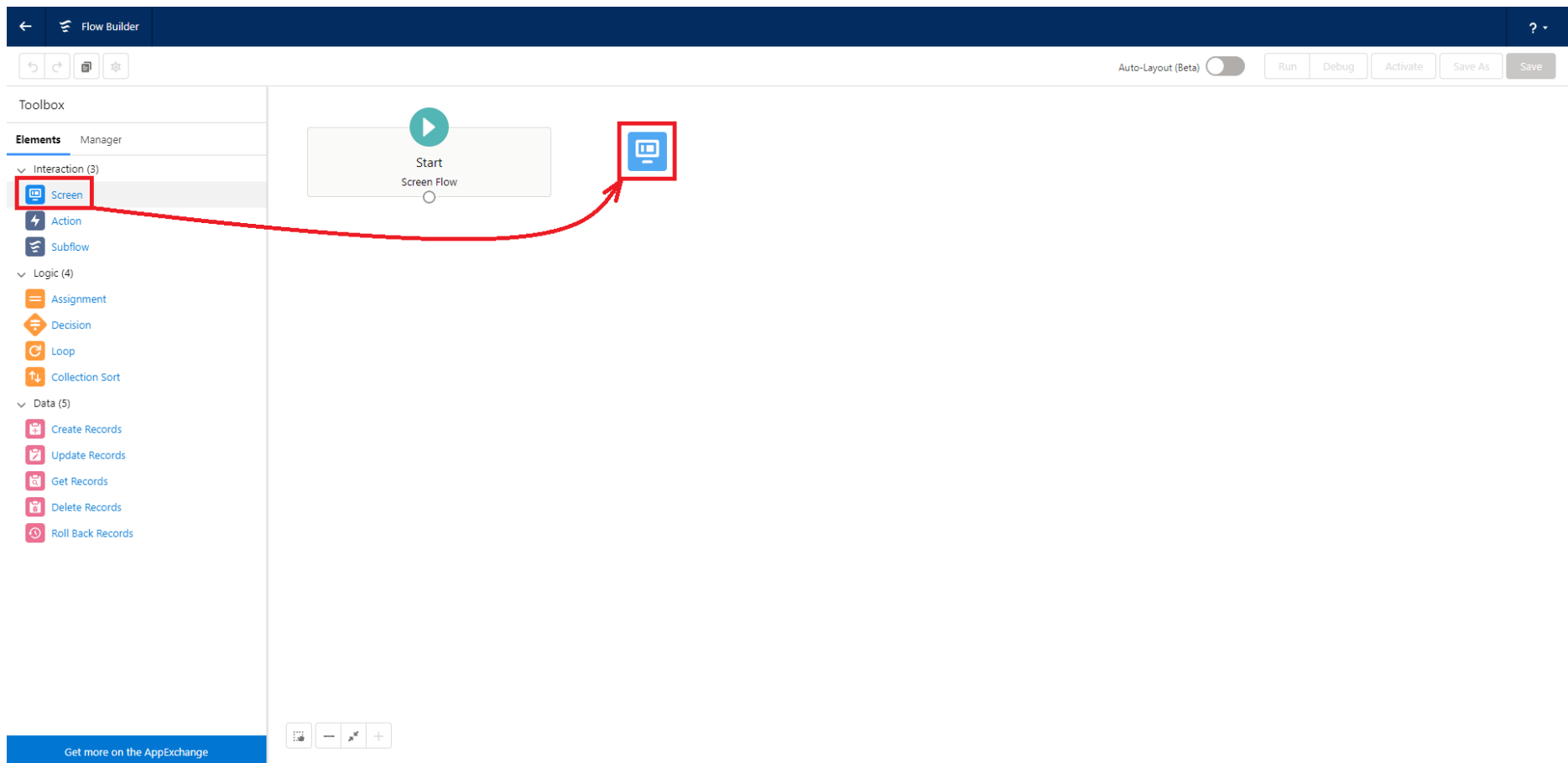
If you create a new Flow, select Screen Flow type and click Next



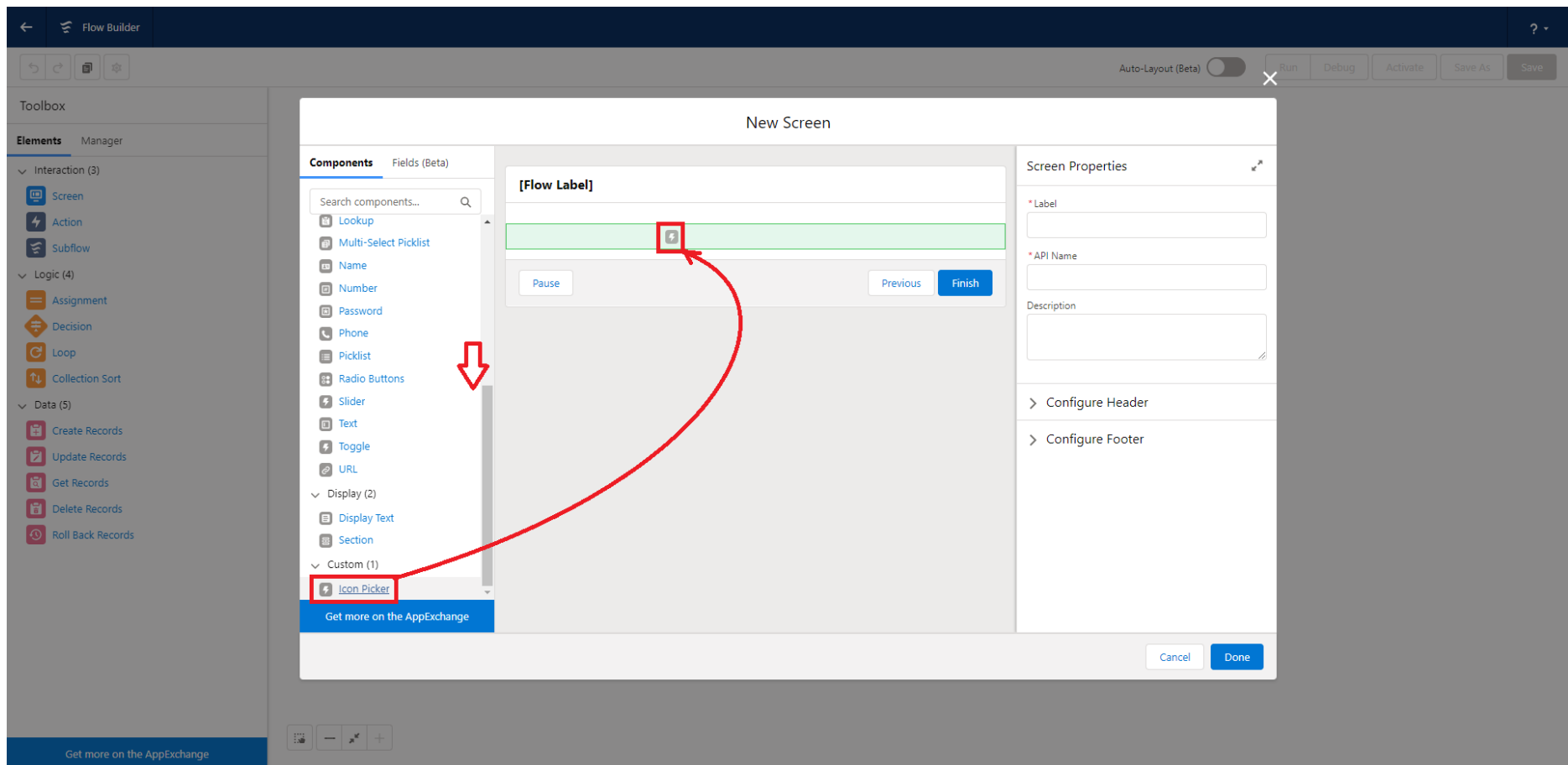
Select Free Form



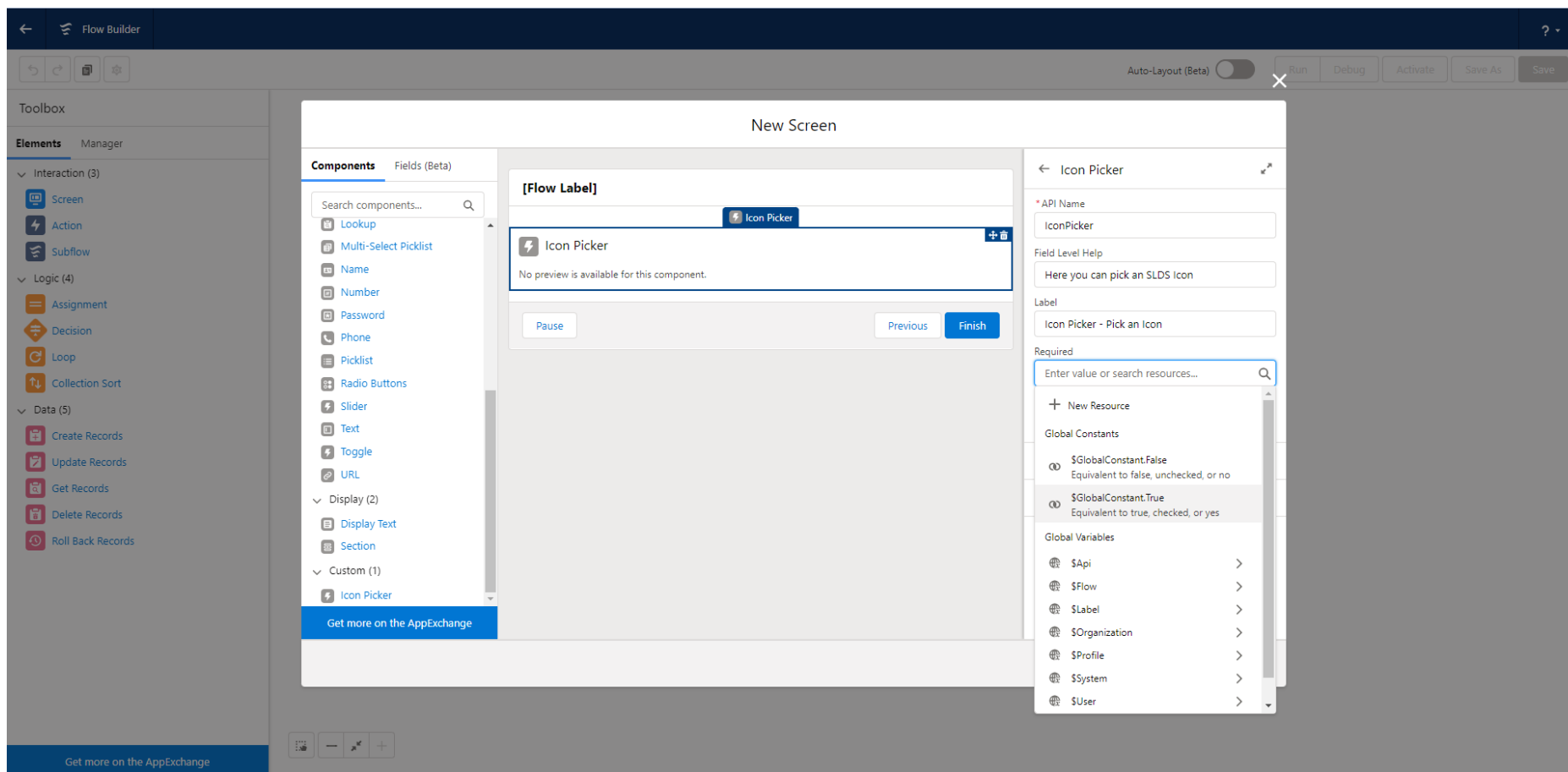
Find Screen in list of Elements on the Elements sidebar panel and drag and drop it into canvas



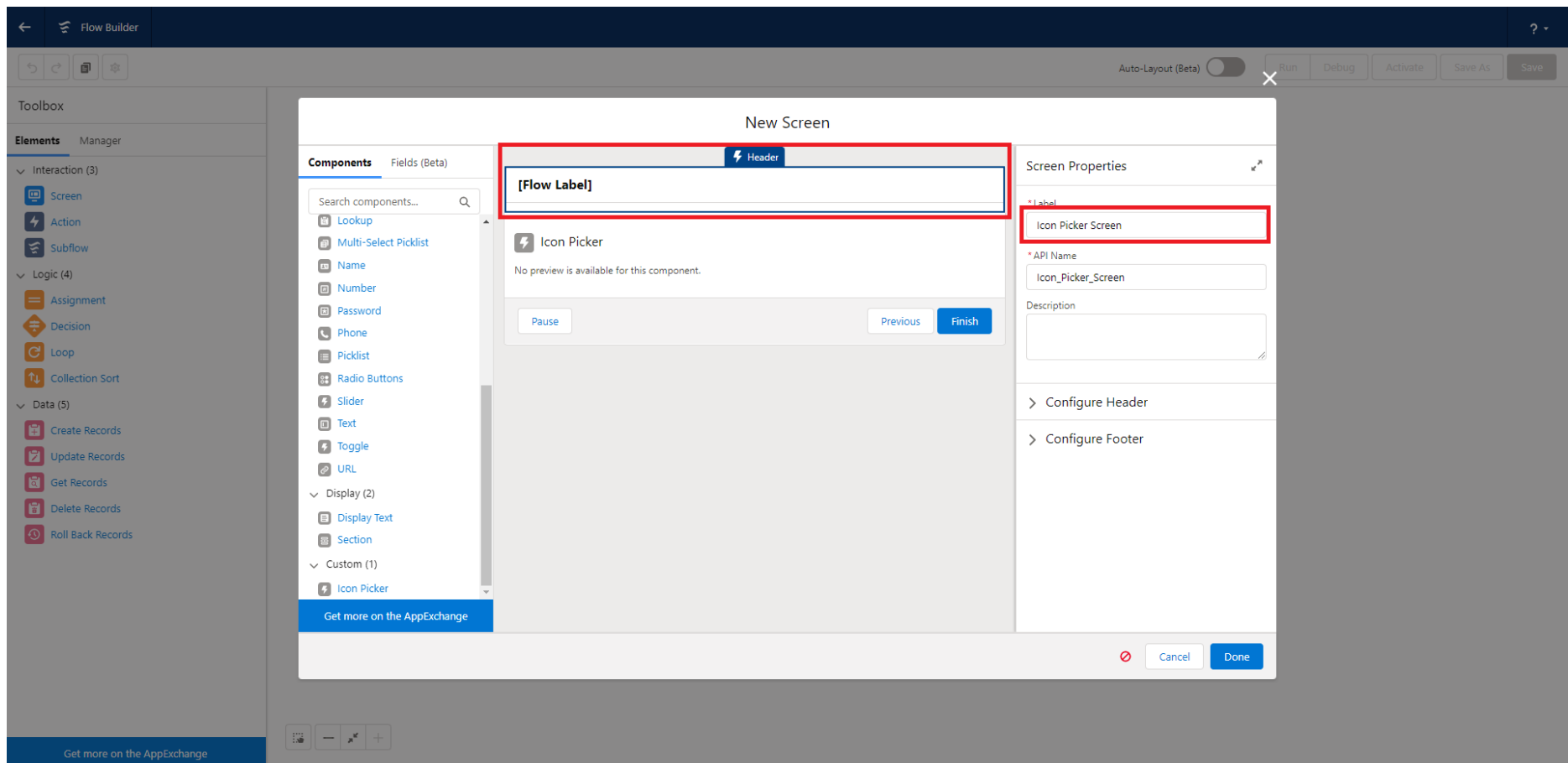
Once you have added Screen to Canvas, scroll down to the last Custom section and select Icon Picker and drop it into Screen canvas.



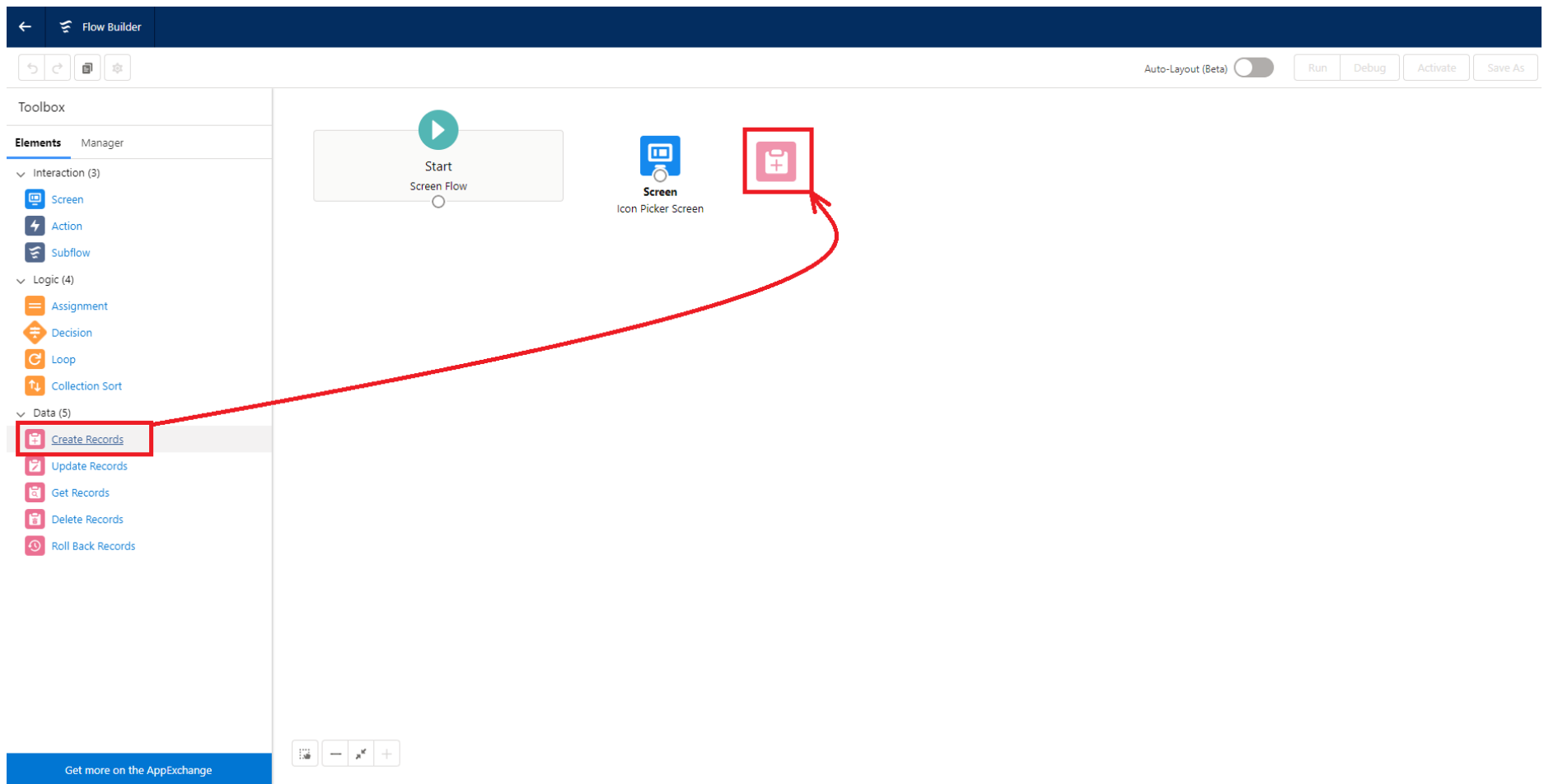
In the sidebar pane, you can set header label and field level help and mandatory indicator, for example, use `Icon Picker - Pick an Icon` like header label and `Here you can pick an SLDS Icon` as field level help. Use `GlobalConstant.true` or `GlobalConstant.false` to show or hide the red asterisk to appear on the screen. Set `IconPicker` as API Name.



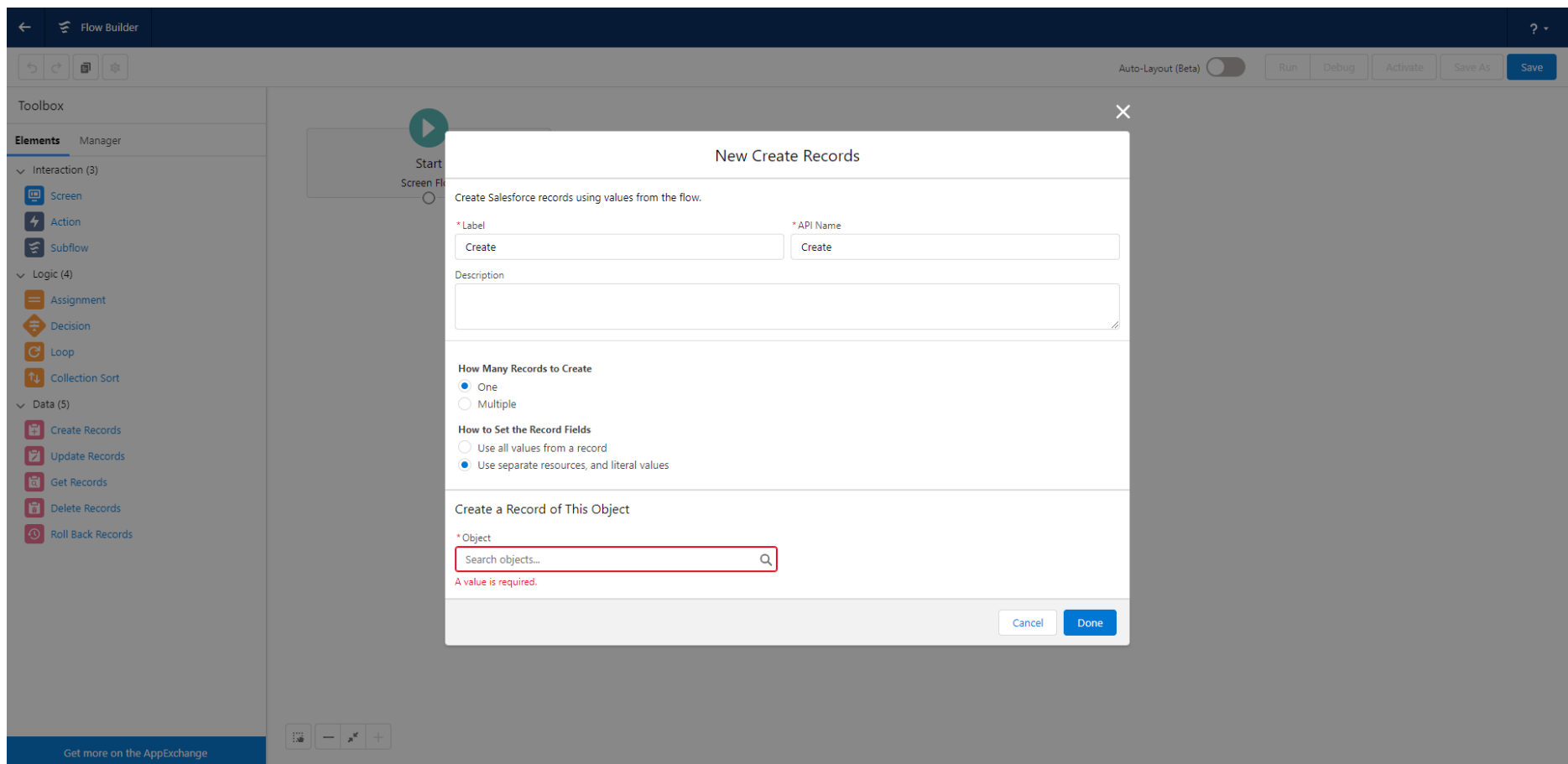
Select screen properties and set value of label to something like `Icon Picker Screen` and API name or keep the default API name. Then click the Done button.



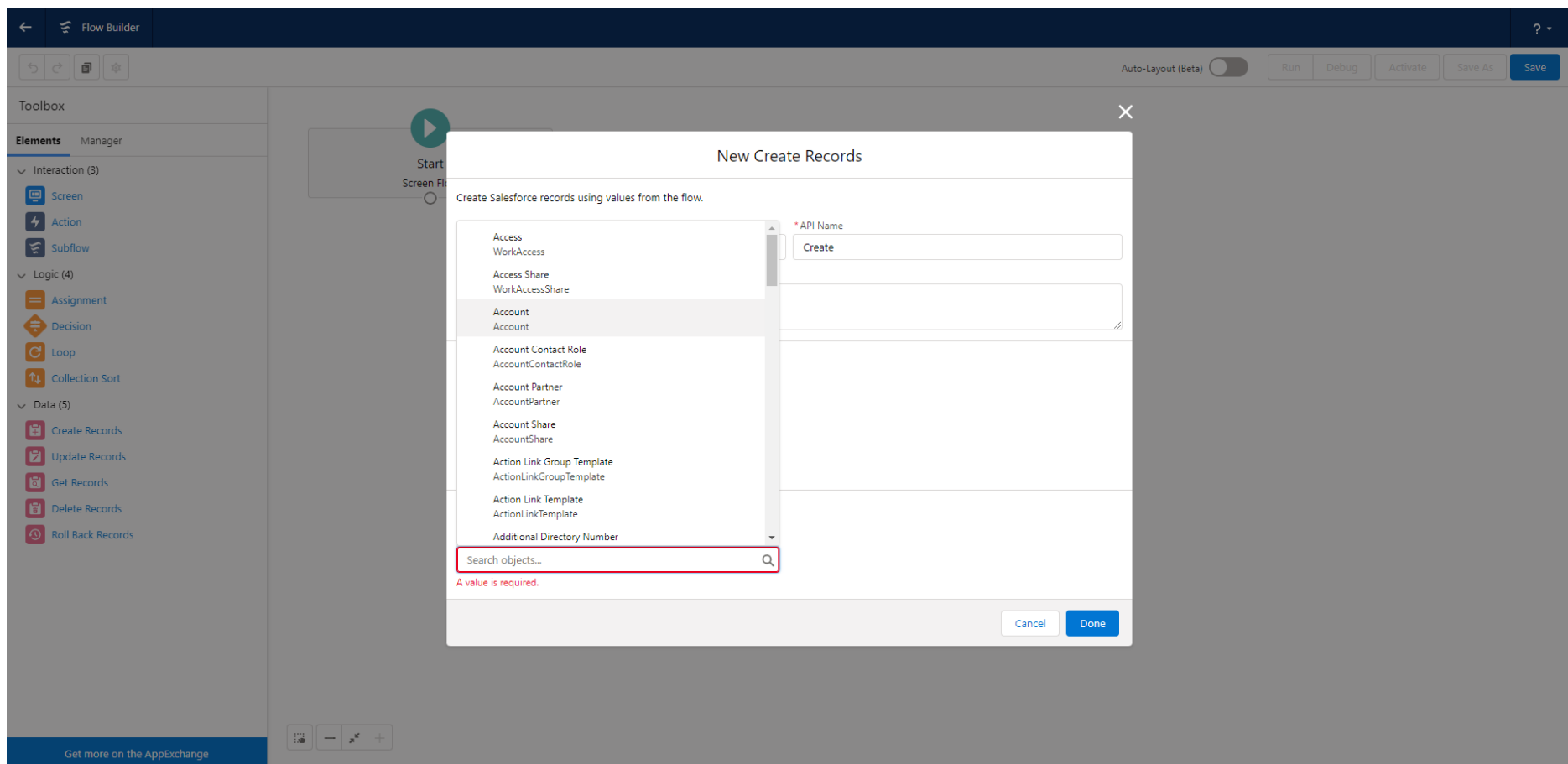
Choose what to do with the value, for example, you can create an Account record, whose Name is icon fullname. For that, select Create Records and drag and drop it into Flow canvas



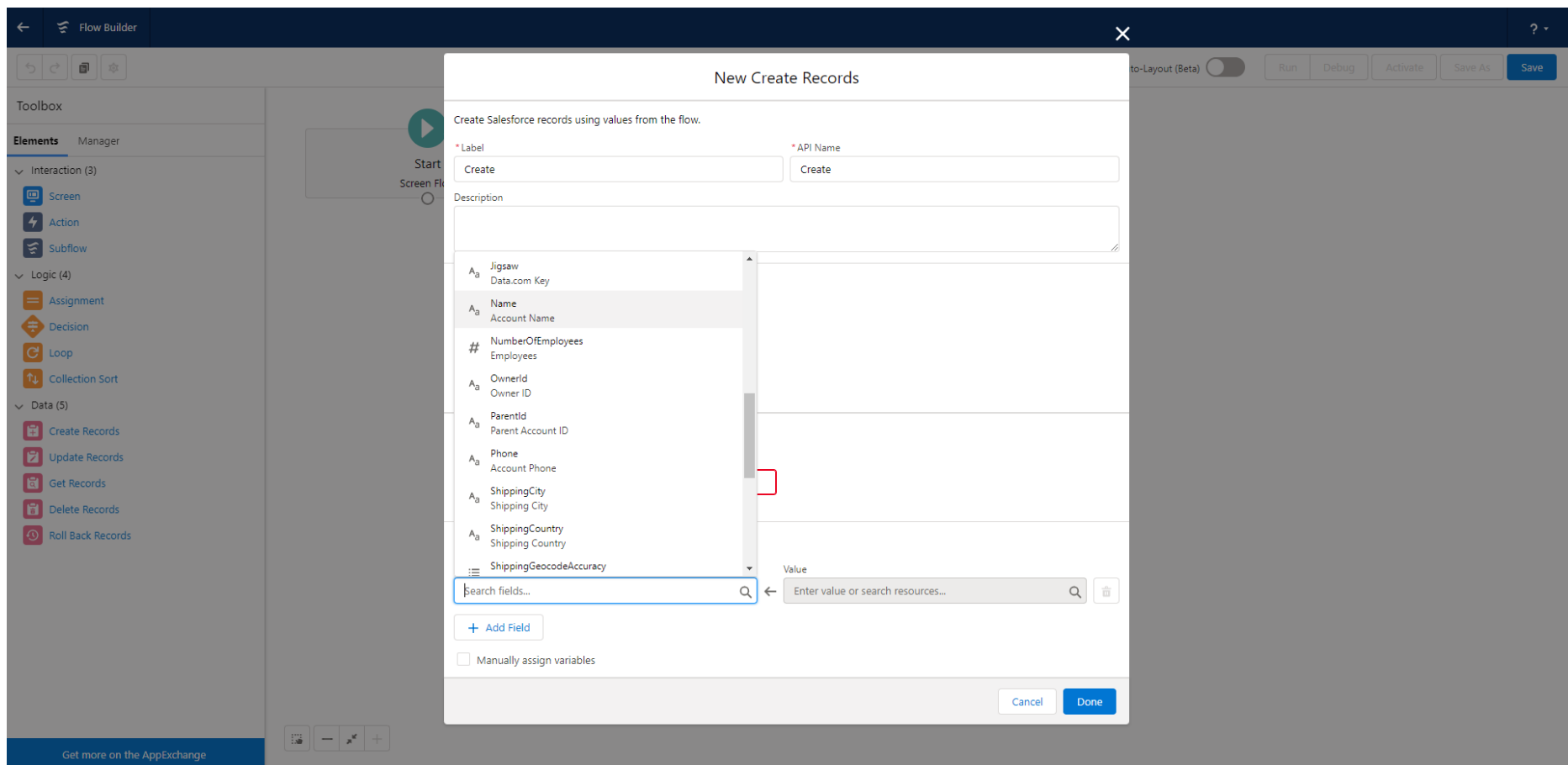
Set Label and API Name to Create, select Use separate resources, and literal values in **How to Set the Record Fields**



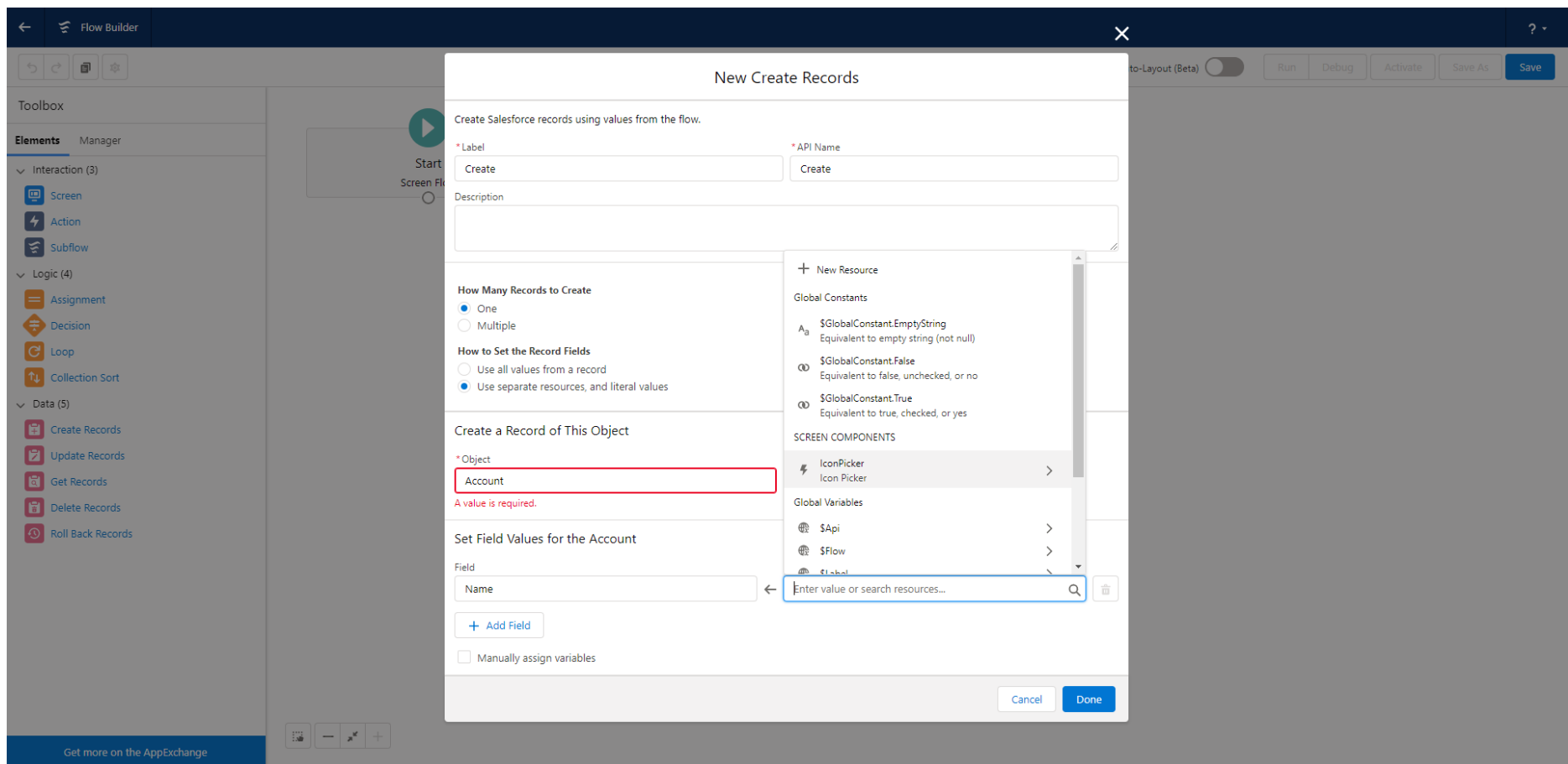
Select Account as Object

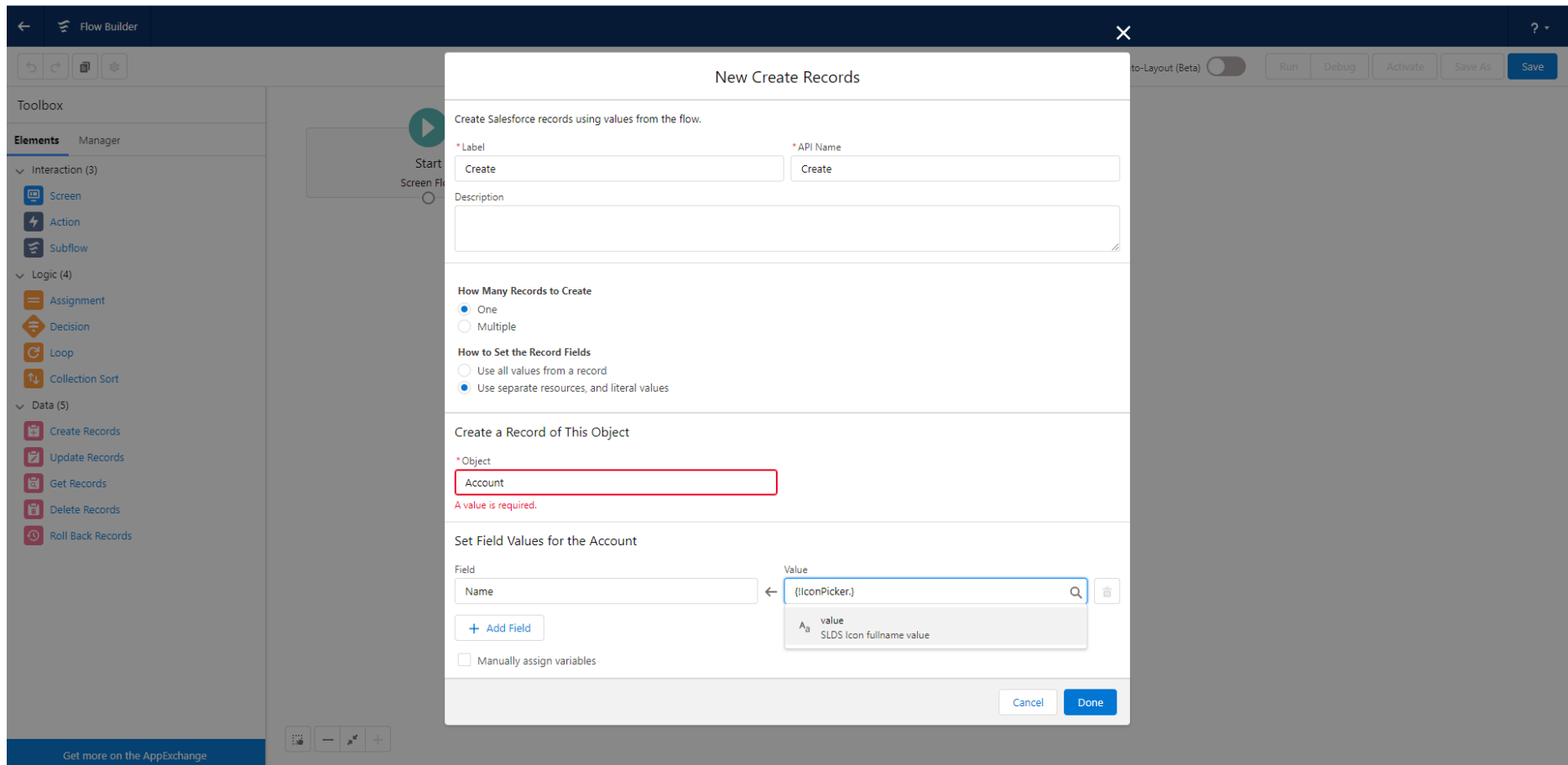


Select Account Name as Field name

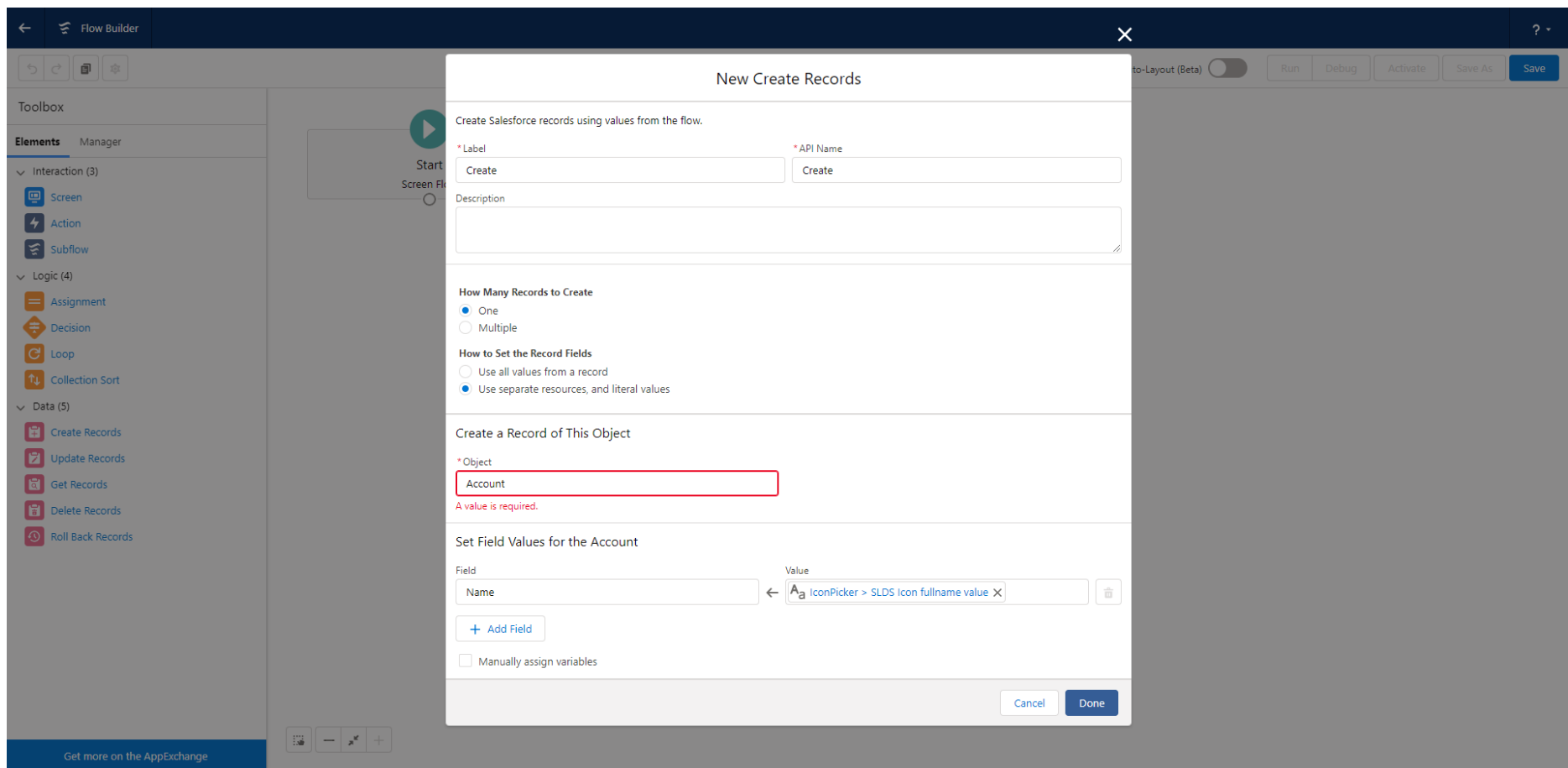


and then in value select Icon Picker and then SLDS Icon Fullname Value

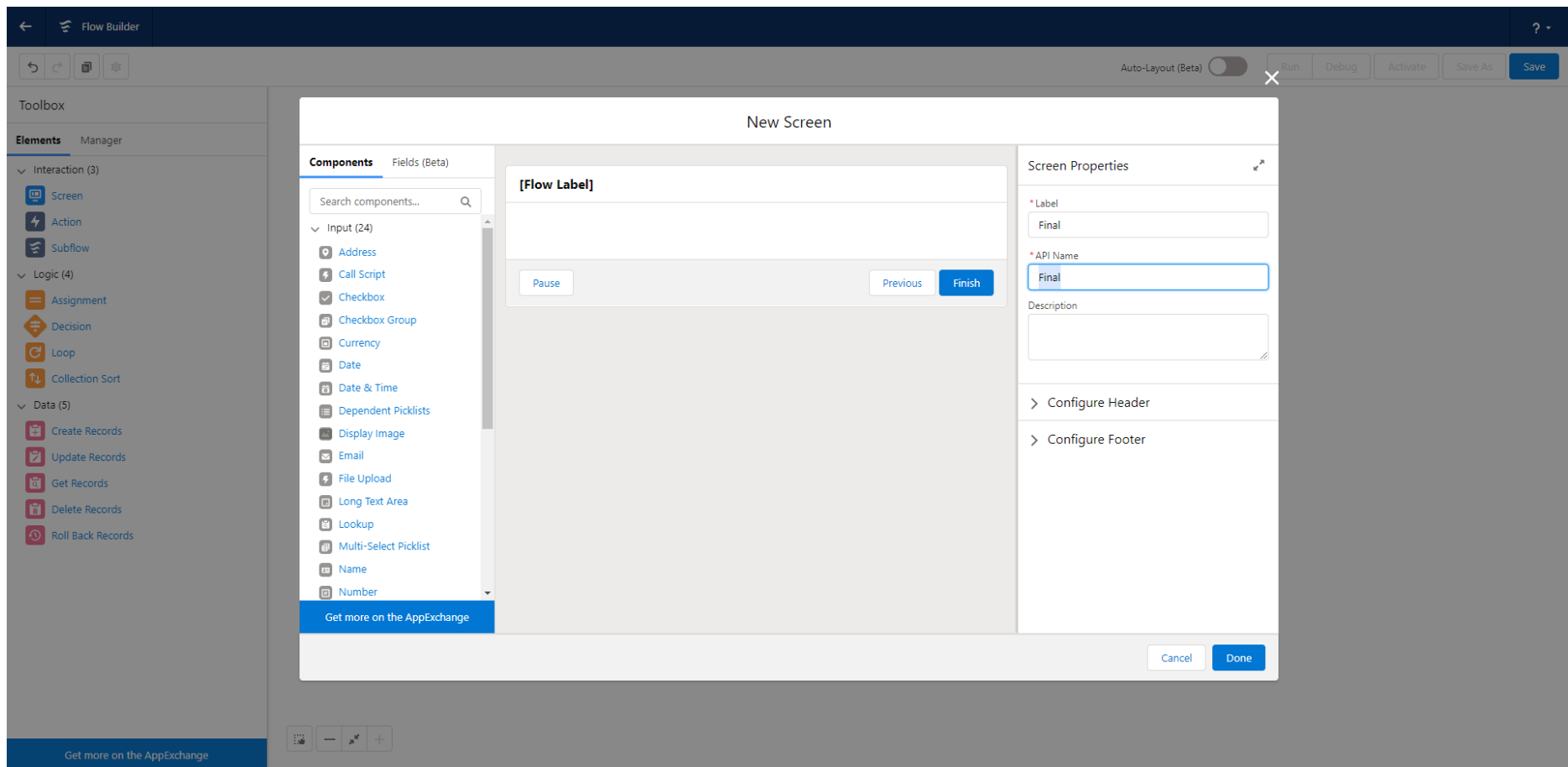




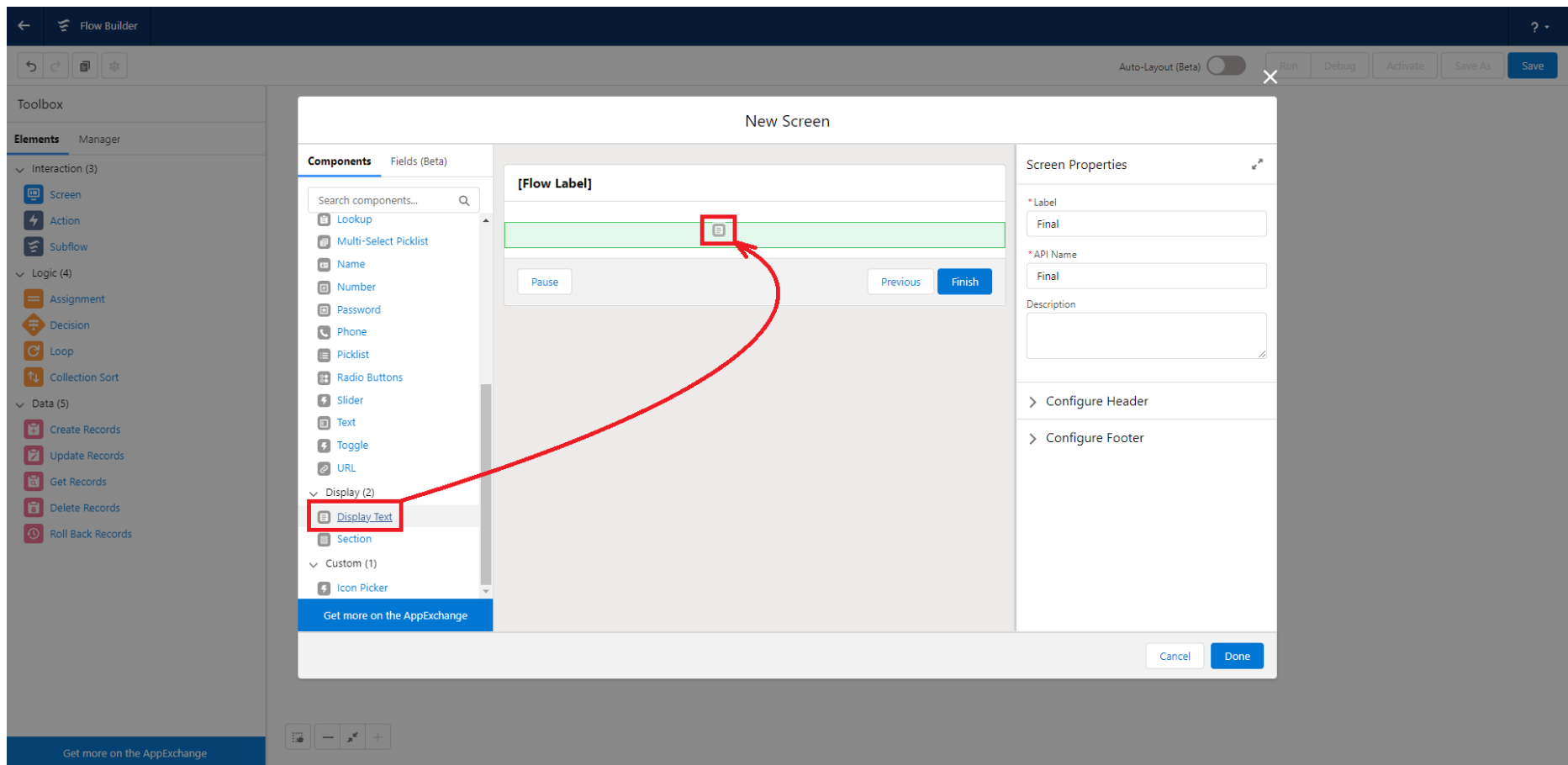
Click on Done once finished



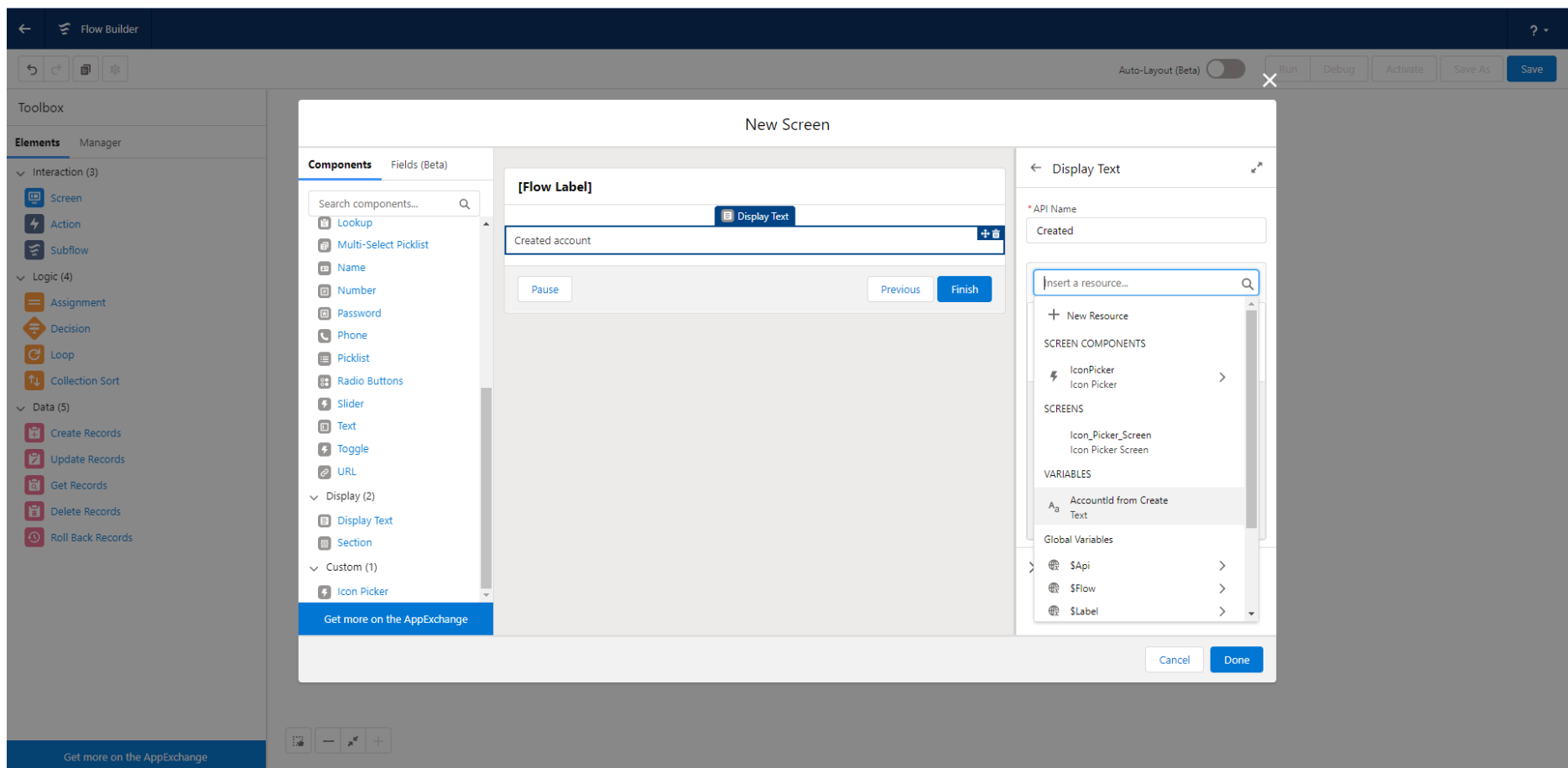
Add another Screen element and set Label and API Name to Final



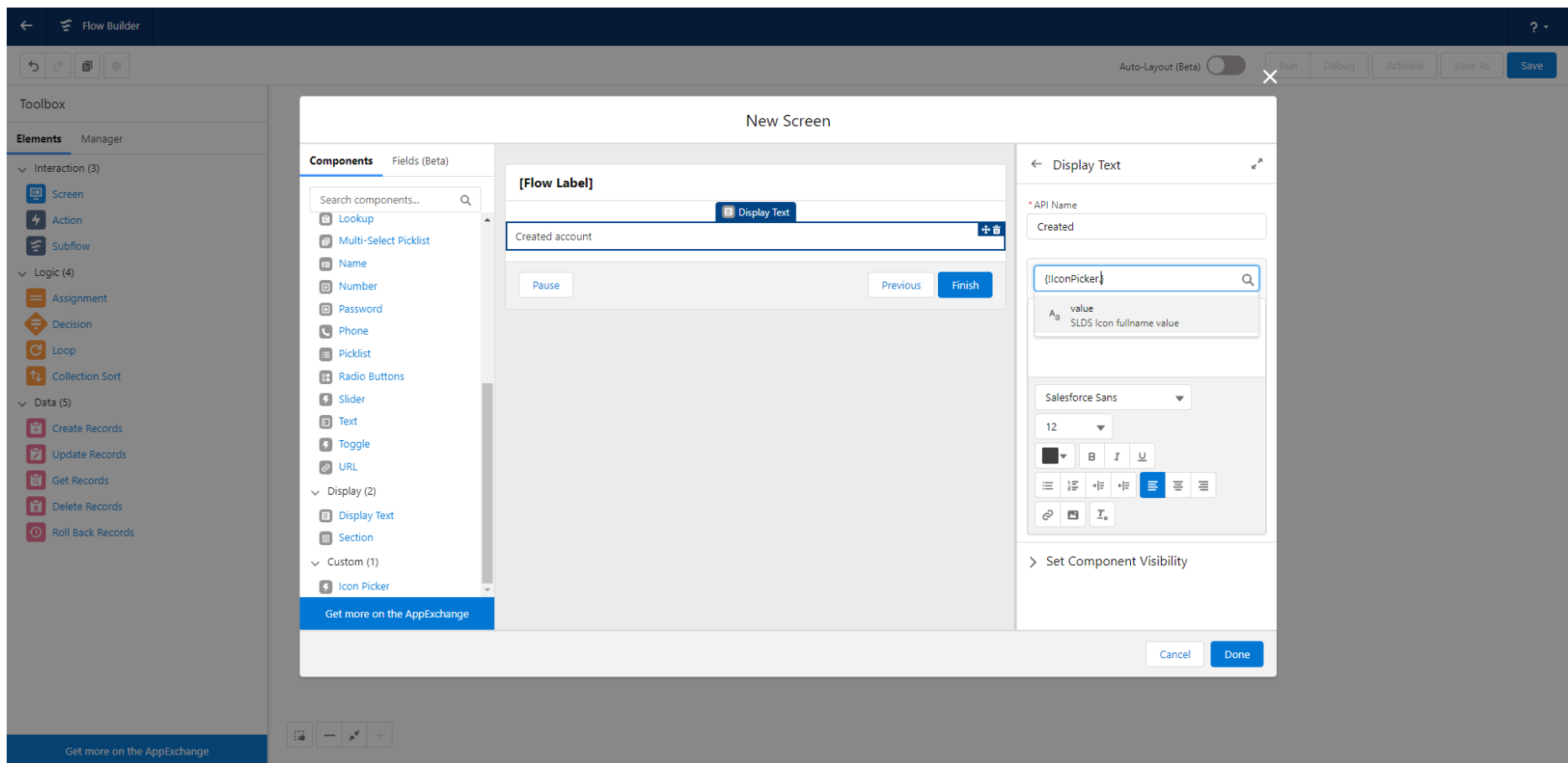
Add a Display Text component to the Final Screen



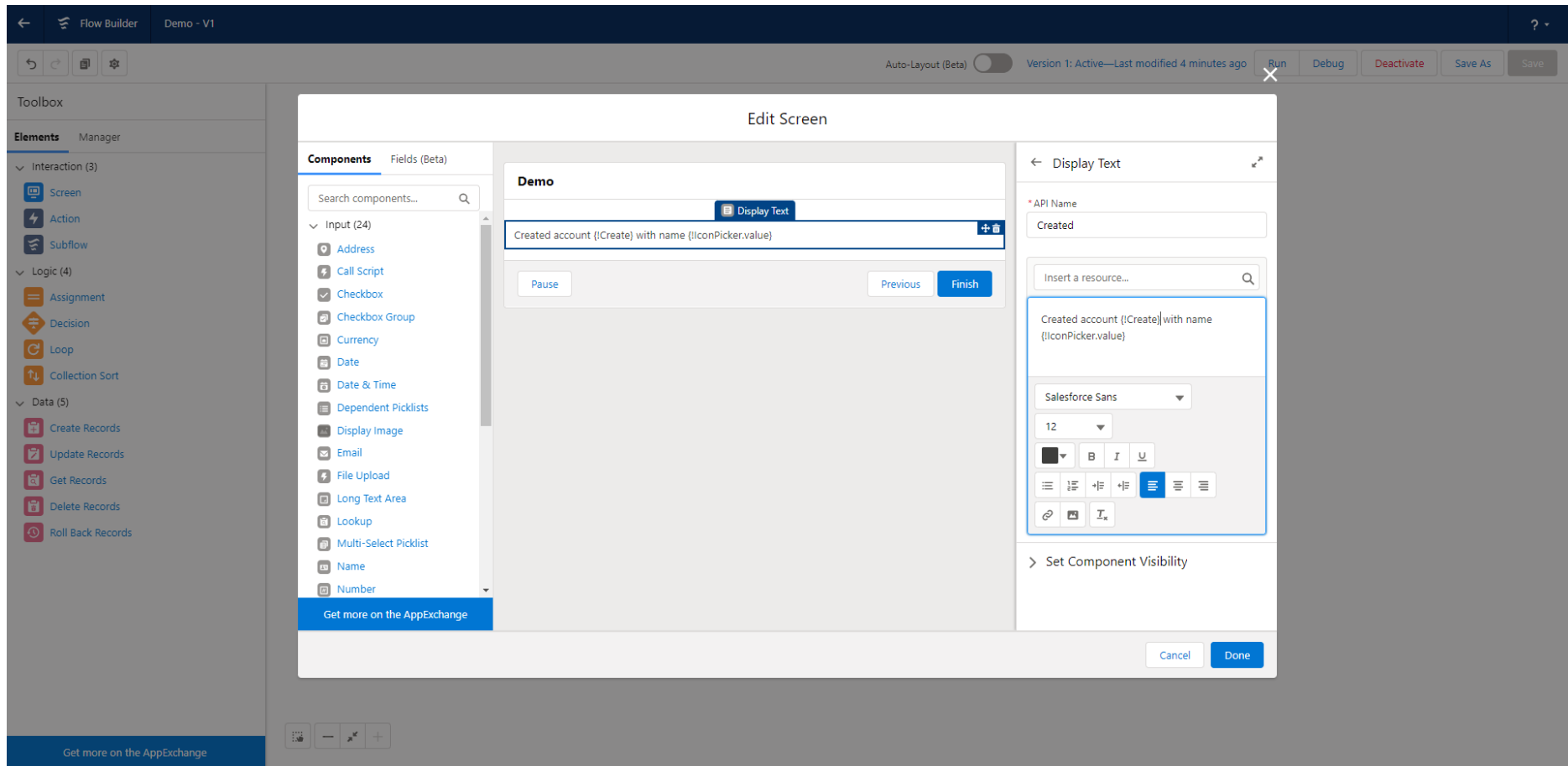
Set API name to Created and Set Created account into the Text Box and then click insert Resource and select AccountId from Create



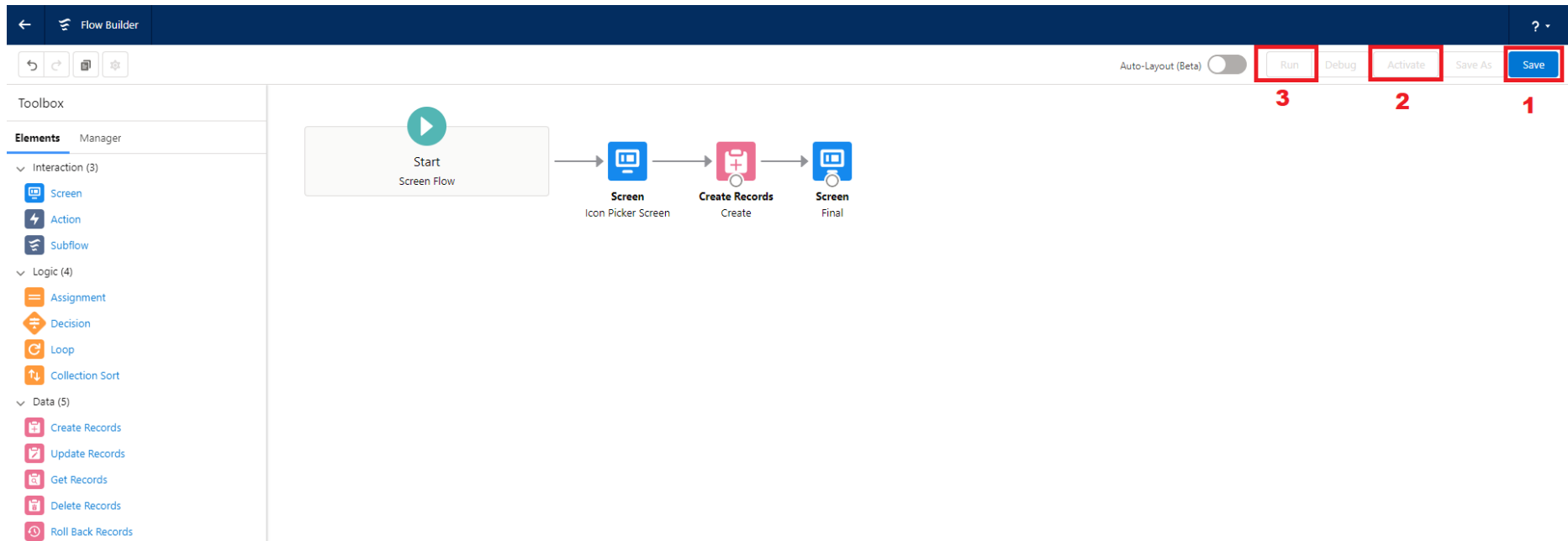
Then add with name and click again Insert a resource and select Icon Picker and then select Value (SLDS Icon Fullname Value)



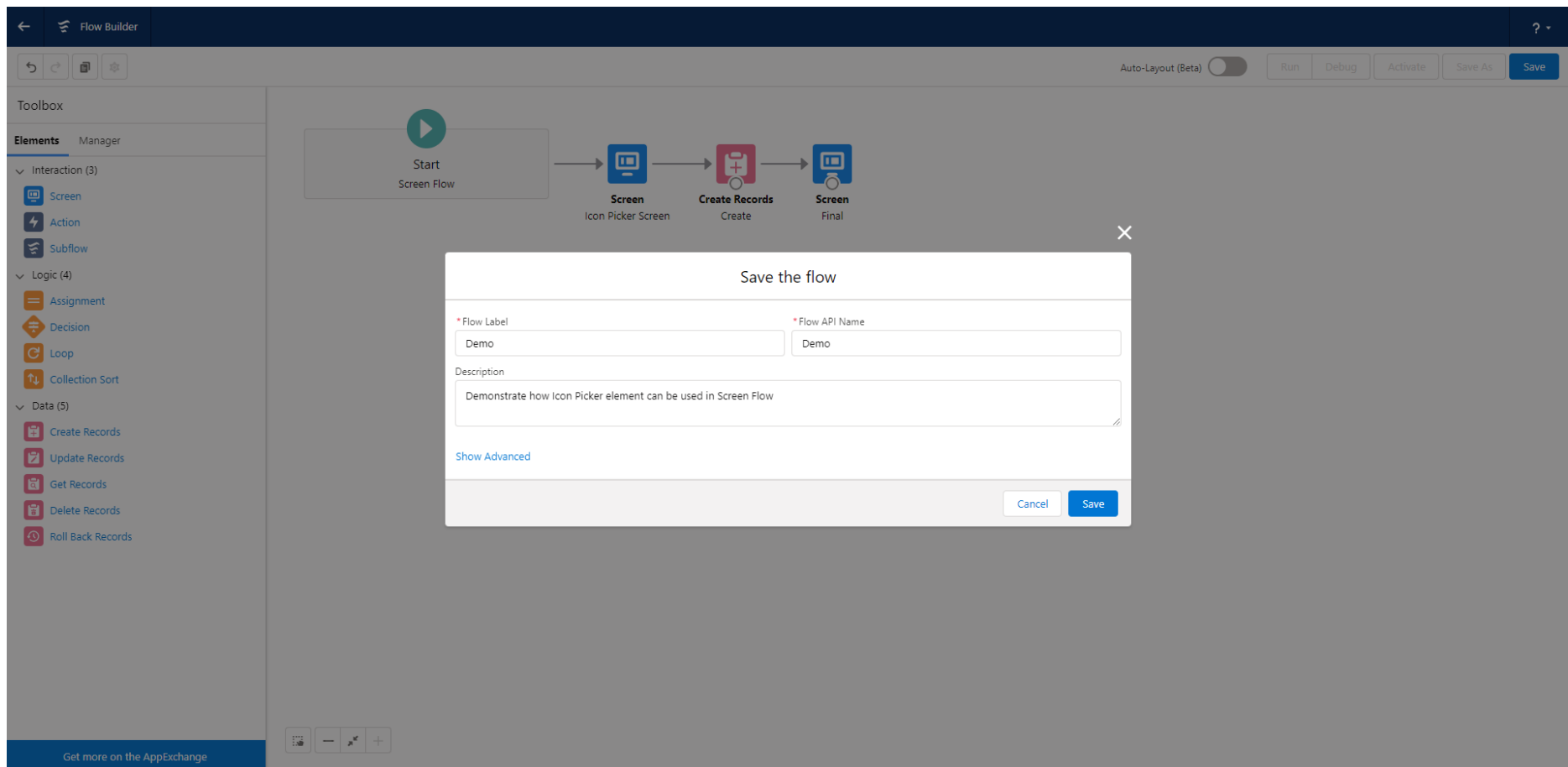
Click Done now



Then connect Start element to Screen element and then Screen Element to Create records component and Create records component to Final Screen element.



Now click Save, then Activate and then Run. When prompted, set label and API name of Flow to Demo and set Description to `Demonstrate how Icon Picker element can be used in Screen Flow` and click Save



Once saved, click Activate and Run

←

Flow Builder

Demo - V1

?

↶ ↷ 📄 ⚙

Auto-Layout (Beta) ☐ Version 1: Inactive—Last modified a few seconds ago

Run Debug Activate Save As Save

Toolbox

Elements

Manager

Interaction (3)

Screen

Action

Subflow

Logic (4)

Assignment

Decision

Loop

Collection Sort

Data (5)

Create Records

Update Records

Get Records

Delete Records

Roll Back Records

Get more on the AppExchange

▶

Start

Screen Flow

→

📺

Screen

Icon Picker Screen

→

📄

Create Records

Create

→

📺

Screen

Final

⋮ − 🔍 +

When you run the demo, pick some icon

Demo

* Icon Picker - Pick an Icon ⓘ

Empty

- apex_plugin (standard)
- app (standard)
- approval (standard)
- apps (standard)

and then click Next

Demo

* Icon Picker - Pick an Icon ⓘ

approval (standard)

Next

Now you should be able to see the confirmation that a new Account with given name which is equal to Icon name was created and now you can click the Finish button.

Demo

Created account 0018H0000004LXD1QAO with name standard:approval

Previous Finish

Using IconPicker in Custom LWC Development

You can use this component in your Custom LWC Development. Before you start, go to Setup Session Setting and check checkbox Use Lightning Web Security (beta) for Lightning web components. and click the Save button.

The screenshot shows the Salesforce Setup interface. The left sidebar contains a search bar with 'session' entered, and a navigation menu with 'Security', 'Session Management', and 'Session Settings' (highlighted). The main content area is titled 'Session Settings'. It includes a section for 'Embedded Service' with a note about enabling it by adding external domains. Below this are several security settings sections: 'Trusted Domains' (with an 'Add Domain' button), 'Cross-Site Request Forgery (CSRF) Protection' (with checkboxes for GET and POST requests), 'Content Security Policy protection' (with checkboxes for overriding restrictions and enabling stricter policy), 'Lightning Locker API Version' (with a dropdown set to 53.0), 'Lightning Web Security (Beta)' (with a checked checkbox for using LWS for LWC components, highlighted by a red box), 'XSS protection' (with a checked checkbox), 'Content Sniffing protection' (with a checked checkbox), 'Referrer URL protection' (with a checked checkbox), 'Redirections' (with a checked checkbox), and 'Session Security Levels'.

If you don't perform this step, you will not be able to use cross-namespaced LWC components in your development. Then create a new LWC and put some code into html file like this

```
<template>
  <iconPicker-icon-picker label="Picker" value={icon} onselect={handleSelect}></iconPicker-icon-picker>

  Label: {selectedItem.label} <br/>
  Icon fullname: {selectedItem.icon}
```

```
</template>
```

Go then to javascript file and put code like this

```
import { LightningElement, track, api } from 'lwc';

export default class TestComp extends LightningElement {
  @api icon;
  @track selectedItem = {};
  handleSelect(event) {
    this.selectedItem = event.detail;
  }
}
```

Go to js-meta.xml metadata file and put code like this

```
<?xml version="1.0" encoding="UTF-8"?>
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
  <apiVersion>53.0</apiVersion>
  <isExposed>true</isExposed>
  <targets>
    <target>lightning__Tab</target>
    <target>lightning__RecordPage</target>
    <target>lightning__AppPage</target>
    <target>lightning__HomePage</target>
  </targets>
  <targetConfigs>
    <targetConfig targets="lightning__AppPage">
      <property name="icon" label="SLDS Icon fullname value" type="String" />
    </targetConfig>
  </targetConfigs>
</LightningComponentBundle>
```

Setting target configuration will allow you to use your custom component on the App page using App Builder. If you want to [put your custom LWC component or Icon Picker to App page using App Builder, refer to the first section](#). If you want to read more about [global attributes and methods of Icon Picker component, refer to the next section](#)

Global attributes and methods of Icon Picker component

Before you start using any of attributes or methods, make sure you have read the previous section and have enabled Use Lightning Web Security (beta) for Lightning web components in Setup\Session Settings.

Attributes

NAME	TYPE	ACCESS	REQUIRED	DEFAULT	DESCRIPTION
header	String	global			The header label for the icon picker input
mandatory	Boolean	global		false	The indicator for the red box requiredness option
fieldLevelHelp	String	global			The help tooltip displayed on hover over the information icon when set
value	String	global			The value of selected icon (input and output)

Methods

NAME	ARGUMENTS	ACCESS	DESCRIPTION
reportValidity		global	Displays the error messages and returns false if the input is invalid. If the input is valid, reportValidity() clears displayed error messages and returns true.

Date of this document: 2nd January of 2022
Version of the Icon Picker component: IP 0.0.0