LeanData

One-Time Routing API Implementation Guide

Overview

Route records through LeanData's One-Time Routing with a click of a button by invoking a LeanDataAPI method. This feature allows users to leverage LeanData API functionality (without writing code to invoke the functions with REST calls).

Example: Reps can route an individual Lead through LeanData 1x Routing without access to the LeanData app.

Please Note: This guide is designed for someone with advanced Salesforce administration skills and experience. It is not intended for beginner level users.

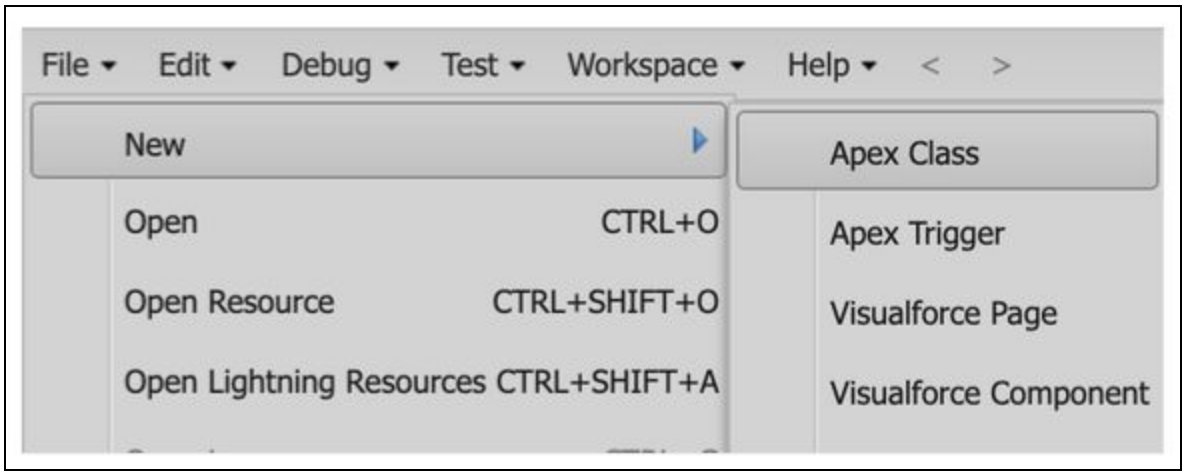
Walkthrough

In this guide, we'll walk through the steps to create a Lightning Action Button on the Lead View that runs the specific Lead through a One-Time Routing Graph using the One-TimeRouting API. There are 3 steps to implement this functionality:

1. Create an Apex Class for Invocable Methods.
2. Create a Flow.
3. Create a Lightning Action Button. For this walkthrough, we will be using the Lightning Experience.

Create an APEX class for Invocable Methods

The first step is to create an *@InvocableMethod* to use in the flow for Step 2. To create a class, navigate to the Developer Console and create a new Apex File. Creating a new class within the Developer Console automatically creates the metadata file needed.



Invocable Method

The Lightning Action Button calls the API. Since we are using the screens in a Flow to capture the input, we need a class to receive them. This class will be the middleman that receives input from the Flow, and calls methods in the *LeanDataAPI.cls* with the input. You can learn more about Invocable Methods here.

Add LeanDataInvocableMethods.cls

In this class, we have created a local class *OneTimeRoutingFlowInputs* in order to encapsulate all the input variables that we wish to take in from the flow. The variables are decorated with an *@InvocableVariable* tag so that the flow can idenfy these variables.

In this example, we've decided to create 4 inputs so we have the flexibility to use this class for any record of any object, and route it through any node type of any graph.

```
public class LeanDataInvocableMethods{
  //input details that comes to apex from flow
  public class OneTimeRoutingFlowInputs{
        @InvocableVariable
        public String sObjectType;    @InvocableVariable
        public Id sObjectId;    @InvocableVariable
        public String graphName;    @InvocableVariable
        public String nodeType;
} @InvocableMethod(label='LeanData - One Time Routing' description='Call
public static void invocableoneTimeRouting(List<OneTimeRoutingFlowInputs> i
        Map<String, Object> paramsMap = new Map<String, Object>();
        paramsMap.put('objectType', inputs[0].sObjectType);
```

```
        paramsMap.put('condition', 'Id = \'' + inputs[0].sObjectId+'\'');
        paramsMap.put('graphName', inputs[0].graphName);
        paramsMap.put('nodeType', inputs[0].nodeType);
        paramsMap.put('notificationsDisabled', true);
        paramsMap.put('sendEmail', true);
        paramsMap.put('allowDedupe', false); if(!Test.isRunningTest())
                LeanDataAPI.oneTimeRouting(paramsMap);
        }
}
```

Test Case

We need to write tests in order to get coverage. Since the execution of LeanData's API is wrapped in a *!Test.isRunningTest()* block, the test method can be as simple as ensuring that the lines before run.

```
@isTest
public static void testInvocableOneTimeRouting() {
    Lead testLead = new Lead(LastName = 'LeadTest', Company = 'testCompany');
    Insert testLead;
    LeanDataInvocableMethods.OneTimeRoutingFlowInputs testInput = new LeanData
    testInput.sObjectType = 'testObj';
    testInput.sObjectId = testLead.Id;
    testInput.graphName = 'testGraph';
    testInput.nodeType = 'testNodeType';
    LeanDataInvocableMethods.invocableoneTimeRouting(new
    List<LeanDataInvocableMethods.OneTimeRoutingFlowInputs> {testInput});
}
```

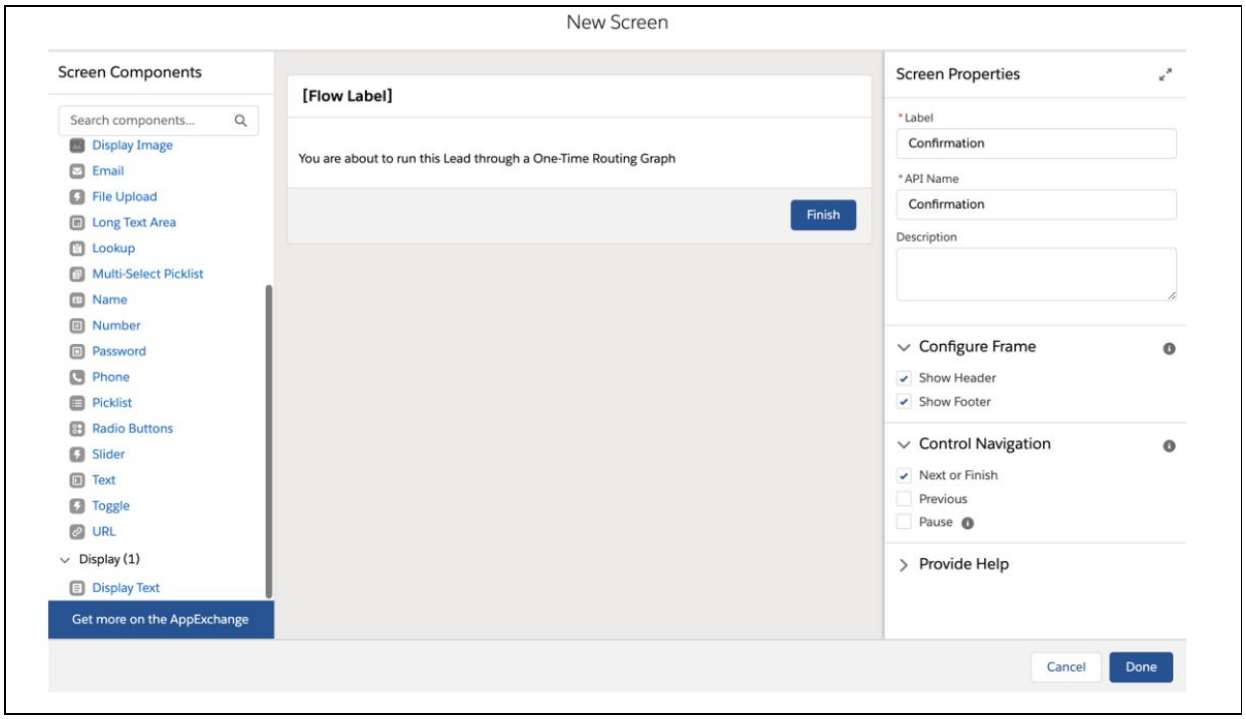
Create a Flow

Now that we have the *@InvocableMethod*, we can create the flow. This flow will be a straightforward flow that consists of 3 components:

1. Confirmation Screen
2. APEX Action to invoke the Invocable Method
3. Success Screen

Confirmation Screen

1. Go to Setup.
2. Search for *Process Automation* in the Quick Find Box on the left panel and select *Flows*.
3. Click on *New Flow*, select *Screen Flow,* then hit the Create button. Screen Flows allow you to take in input to run processes dynamically.
4. From the Elements tab on the left, drag and drop a *Screen* onto the canvas.
5. Double click the *Screen* component to edit it.
6. On the left of the edit screen, we will use a *Display Text* component to display the confirmation message. Drag and drop it onto the layout. Click on it to modify the text content and its API Name.
7. There should be a default footer component on the layout. Click on it and go to the Control Navigation tab. Uncheck Previous and Pause.
8. Enter a Label and API Name in the Screen Properties.
9. Connect the Start and Screen elements.

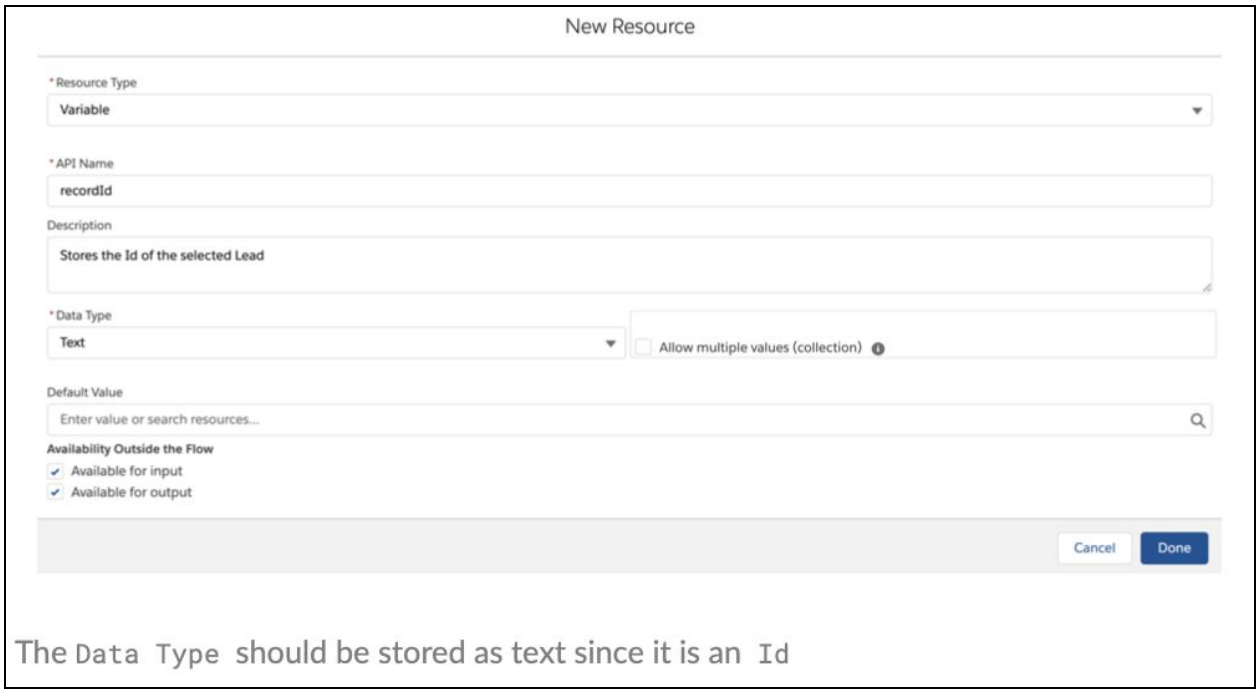


APEX Action

Create Variable Input

We identified the 4 *@InvocableVariable* in the last section. We will need to create variables in the flow to store them.

1. From the *Manager* tab in the Toolbox panel, click on *New Resource* and select *Variable*.
2. Create the first Variable with the details below.
    - Resource Type - Variable
    - API Name - recordID
    - Description - Stores the ID of the Selected Lead
    - Date Type - Text
    - Availability Outside the Flow - Check Available for input and output



Create Constant Input

In this case, we are storing the *objectType, graphName* and *nodeType* as constants since this flow should be designed to do one specific task, to recycle Leads. The object will always be a Lead, and we'll always route it through the Update Trigger of the Recycle Leads graph. You can create different buttons that perform different actions by creating a new Flow and changing the constants. For instance, we would create a Flow that recycles Contacts by simply changing graphName without making any changes to the *LeanDataInvocableMethod.cls*.

1. From the Manager tab, click on New Resource
2. The Data Type should correspond to the @InvocableVariable in the LeanDataInvocableMethods class. (In this example they are all Text)
3. Enter the respective values in the Value input
    - The default values for this example:
    - graphName -> [Name of the Graph]
    - objectType ->

- nodeType -> or INSERT TRIGGER (which corresponds to the update/insert node in the graph)

Create the Action

1. From the Elements tab in the Toolbox panel, drag and drop an Action onto the canvas.
2. Click on it, select Type opon in the Filter By dropdown, and select Apex Action.
3. In the Search input, you should be able to see LeanData - One Time Routing we created in the previous section.
4. Input the Label and API Name for you flow.
5. In the Set Input Values section, toggle the radio checkboxes, and include the respective constant/variables created in previous steps.
6. Connect the Confirmation Screen to the created Apex Action.
7. Save this flow.
8. Click on the Activate button to activate the flow.

Success Screen

Now, we'll repeat the process in part a to create a screen that displays the success message.

1. Connect the Apex Acon to the final screen and hit save.

Create the Lightning Action Button

At this point, we have all the functionality in place, and we need a button to trigger the call. The process is:

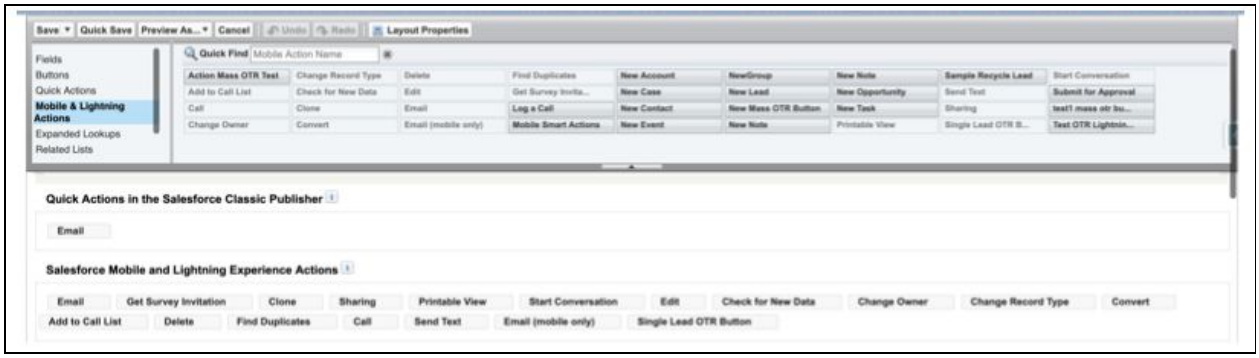
1. Create the button.
2. Add the Action Button to the Page Layout.
3. Test the button.

Create the Action Button

1. Go to Setup.
2. Search for Object Manager in the central search bar and look for Leads in the Quick Find search bar.
3. Click on Buttons, Links, and Actions on the left panel.
4. Click New Action on the top right of the view panel to create the action button.
5. Select Flow for Action Type and pick the flow that was created previously. Enter the name and descriptions.
6. Save.

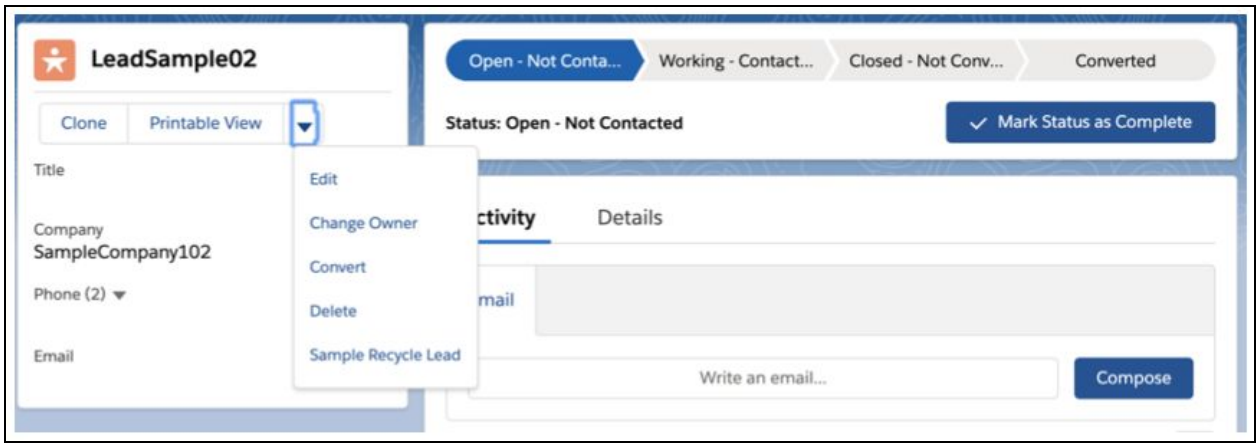
Add the Action Button to the Page Layout

1. Select Page Layouts in the left panel and select Lead Layout.
2. Search for Mobile & Lightning Experience Actions section.
3. Save.



Test the Button

Navigate to a Lead and click on the dropdown to reveal the actions. You should see Sample Recycle Lead button, which on click, will execute one-time routing for this Lead through the Recycle Lead graph.



Summary

This guide has provided an overview on implementing routing of records through LeanData's One-Time Routing with the click of a button by invoking a LeanData API method. This will allow you to leverage the LeanData API without having to write code.

If you have questions or need help please contact integrations@leandatainc.com.