

Flow + Apex + LWC

A Perfect Combination!

Andrés Pérez
Senior Manager,
Trailhead Academy Solution Architect Lead
@ELTOROIT



Andrés Pérez
[ELTOROIT](#)
Senior Manager,
Solution Architect Lead

CERTIFIED Instructor
CERTIFIED System Architect
CERTIFIED Application Architect
CERTIFIED Sharing and Visibility Designer
CERTIFIED Data Architecture and Management Designer
CERTIFIED Identity and Access Management Designer
CERTIFIED Development Lifecycle and Deployment Designer
CERTIFIED Integration Architecture Designer
CERTIFIED JavaScript Developer I
CERTIFIED Platform Developer I
CERTIFIED Platform Developer II
CERTIFIED Platform App Builder
CERTIFIED Administrator
CERTIFIED Advanced Administrator
CERTIFIED Force.com Developer
CERTIFIED Force.com Advanced Developer

TRAILHEAD ACADEMY

Forward Looking Statements



This presentation contains forward-looking statements about, among other things, trend analyses and future events, future financial performance, anticipated growth, industry prospects, environmental, social and governance goals, and the anticipated benefits of acquired companies. The achievement or success of the matters covered by such forward-looking statements involves risks, uncertainties and assumptions. If any such risks or uncertainties materialize or if any of the assumptions prove incorrect, Salesforce's results could differ materially from the results expressed or implied by these forward-looking statements. The risks and uncertainties referred to above include those factors discussed in Salesforce's reports filed from time to time with the Securities and Exchange Commission, including, but not limited to: impact of, and actions we may take in response to, the COVID-19 pandemic, related public health measures and resulting economic downturn and market volatility; our ability to maintain security levels and service performance meeting the expectations of our customers, and the resources and costs required to avoid unanticipated downtime and prevent, detect and remediate performance degradation and security breaches; the expenses associated with our data centers and third-party infrastructure providers; our ability to secure additional data center capacity; our reliance on third-party hardware, software and platform providers; the effect of evolving domestic and foreign government regulations, including those related to the provision of services on the Internet, those related to accessing the Internet, and those addressing data privacy, cross-border data transfers and import and export controls; current and potential litigation involving us or our industry, including litigation involving acquired entities such as Tableau Software, Inc. and Slack Technologies, Inc., and the resolution or settlement thereof; regulatory developments and regulatory investigations involving us or affecting our industry; our ability to successfully introduce new services and product features, including any efforts to expand our services; the success of our strategy of acquiring or making investments in complementary businesses, joint ventures, services, technologies and intellectual property rights; our ability to complete, on a timely basis or at all, announced transactions; our ability to realize the benefits from acquisitions, strategic partnerships, joint ventures and investments, including our July 2021 acquisition of Slack Technologies, Inc., and successfully integrate acquired businesses and technologies; our ability to compete in the markets in which we participate; the success of our business strategy and our plan to build our business, including our strategy to be a leading provider of enterprise cloud computing applications and platforms; our ability to execute our business plans; our ability to continue to grow unearned revenue and remaining performance obligation; the pace of change and innovation in enterprise cloud computing services; the seasonal nature of our sales cycles; our ability to limit customer attrition and costs related to those efforts; the success of our international expansion strategy; the demands on our personnel and infrastructure resulting from significant growth in our customer base and operations, including as a result of acquisitions; our ability to preserve our workplace culture, including as a result of our decisions regarding our current and future office environments or work-from-home policies; our dependency on the development and maintenance of the infrastructure of the Internet; our real estate and office facilities strategy and related costs and uncertainties; fluctuations in, and our ability to predict, our operating results and cash flows; the variability in our results arising from the accounting for term license revenue products; the performance and fair value of our investments in complementary businesses through our strategic investment portfolio; the impact of future gains or losses from our strategic investment portfolio, including gains or losses from overall market conditions that may affect the publicly traded companies within our strategic investment portfolio; our ability to protect our intellectual property rights; our ability to develop our brands; the impact of foreign currency exchange rate and interest rate fluctuations on our results; the valuation of our deferred tax assets and the release of related valuation allowances; the potential availability of additional tax assets in the future; the impact of new accounting pronouncements and tax laws; uncertainties affecting our ability to estimate our tax rate; uncertainties regarding our tax obligations in connection with potential jurisdictional transfers of intellectual property, including the tax rate, the timing of the transfer and the value of such transferred intellectual property; uncertainties regarding the effect of general economic and market conditions; the impact of geopolitical events; uncertainties regarding the impact of expensing stock options and other equity awards; the sufficiency of our capital resources; the ability to execute our Share Repurchase Program; our ability to comply with our debt covenants and lease obligations; the impact of climate change, natural disasters and actual or threatened public health emergencies; and our ability to achieve our aspirations, goals and projections related to our environmental, social and governance initiatives.

Updated: September 28, 2022

Linking Family

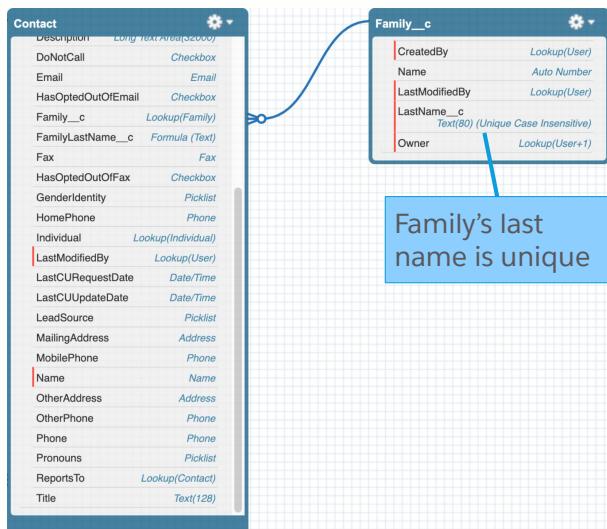
Business requirement for the session



Contacts should be related to their family records, based on their last name.

Whenever a contact is created or updated, find a related family record and link them together.

If a family record is not found, a new family record should be created and linked.



Agenda



- **Exercise 0:** Setup your environment for this session
- **Exercise 1:** Populate family lookup field
- **Exercise 2:** Populate family lookup field (Fixed)
- **Exercise 3:** @invocableMethods with Complex data
- **Exercise 4:** Display contacts grouped by families
- **Exercise 5:** LWC As Custom Property Editors (Homework)
- Q&A

How To Participate In This Session?



3 ways you can enjoy this session...

Answer (Verbal)

Answer (Workbook)

Answer (Workbook) +
Complete exercises

- New to code and flows

- Familiar with code and flows

- Developer who codes Apex and LWC
- Have built complex flows

If you fall behind, just answer the questions ;-)



Exercise 0

Setup your environment for this session



Review: What Are Flows?

Code completion enhanced!



Types Of Flows



New Flow

Core All + Templates

Screen Flow Guides users through a business process that's launched from Lightning pages, Experience Cloud sites, quick actions, and ...	Record-Triggered Flow Launches when a record is created, updated, or deleted. This autolaunched flow runs in the background.
Schedule-Triggered Flow Launches at a specified time and frequency for each record in a batch. This autolaunched flow runs in the background.	Platform Event—Triggered Flow Launches when a platform event message is received. This autolaunched flow runs in the background.
Autolaunched Flow (No Trigger) Launches when invoked by Apex, processes, REST API, and more. This autolaunched flow runs in the background.	Record-Triggered Orchestration Launches when a record is created or updated. An orchestration lets you create a multi-step, multi-user process.

Create

What Is A Record-Triggered Flow?

Automatically executed on DML events



Fast Field Updates

Update fields on the record that triggers the flow to run. This high-performance flow runs *before* the record is saved to the database.



Actions and Related Records

Update any record and perform actions, like send an email. This more flexible flow runs *after* the record is saved to the database.





Record-Triggered Flows

Differences: Record-Trigger Flows vs. Apex Triggers



	Flows	Apex
Stops DML transaction (validations)	No	Yes
Entry condition	Automatic	Manual
Asynchronous path	Automatic	Manual
Undelete Event	No	Yes
How many (per object/event)?	Many	One (Best Practice)
Queries	Simple	Relationships, Aggregates
Bulkification	Automatic	Manual
New record	\$Record (1 record)	Trigger.new (list)
Old Record	\$Record__Prior (1 record)	Trigger.old (list)

Creating A Record-Triggered Flow



which sObject?

Configure Start

Select Object
Select the object whose records trigger the flow when they're created, updated, or deleted.

* Object
Account

Select DML event

Configure Trigger

* Trigger the Flow When:
 A record is created
 A record is updated
 A record is created or updated
 A record is deleted

Are there any entry conditions?

Set Entry Conditions

Specify entry conditions to reduce the number of records that trigger the flow and the number of times the flow is executed. Minimizing unnecessary flow executions helps to conserve your org's resources.

If you create a flow that's triggered when a record is updated, we recommend first defining entry conditions. Then select the **Only when a record is updated** to meet the condition requirements option for When to Run the flow for Updated Records.

Condition Requirements

Formula Evaluates to True

None

All Conditions Are Met (AND)

Any Condition Is Met (OR)

Custom Condition Logic Is Met

▼ Formula Evaluates to True

Check Syntax

Actions... Insert a function... Select an Operator...

Flow type

* Optimize the Flow for:

Fast Field Updates

Update fields on the record that triggers the flow to run. This high-performance flow runs before the record is saved to the database.

Actions and Related Records

Update any record and perform actions, like send an email. This more flexible flow runs after the record is saved to the database.

Need an Async path?

Include a Run Asynchronously path to access an external system after the original transaction for the triggering record is successfully committed

Cancel Done

Exercise 1

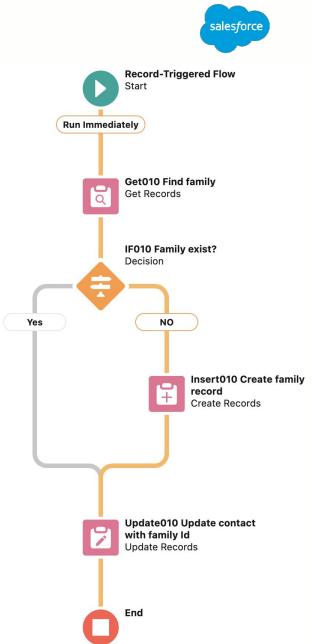
Populate family lookup field



Why Did It Fail?

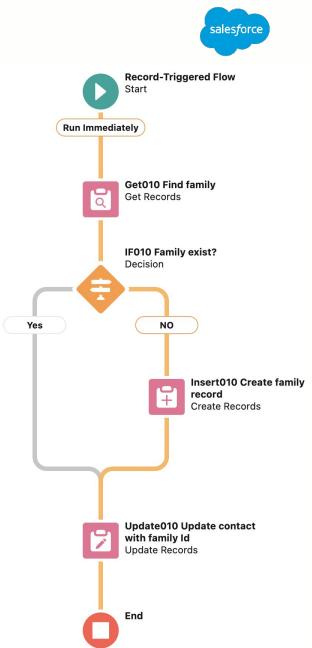
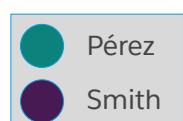
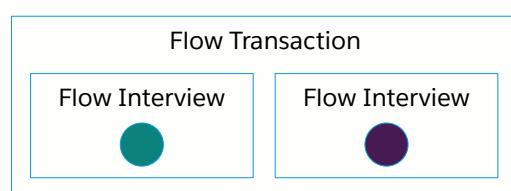
Failed to insert multiple contacts with the same new last name

Why?



Flow Transactions vs. Flow Interviews

Separate but working together!



It Works!

Different new last names

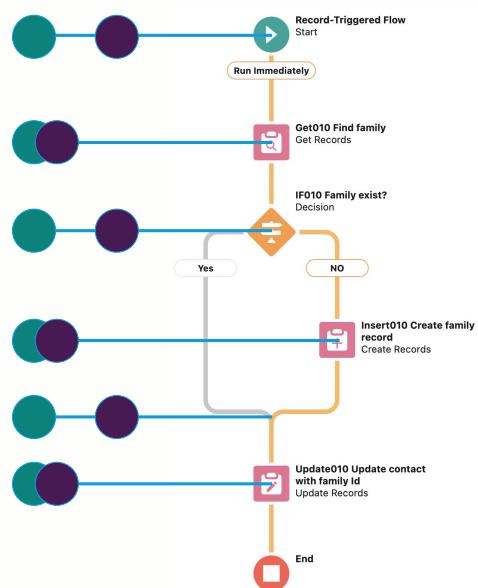
- Bulkification points
 - SOQL / DML elements
 - Screens elements
 - Pause actions
 - Apex calls (optional)

DML Operation



Pérez

Smith



It Fails!

Same new last names

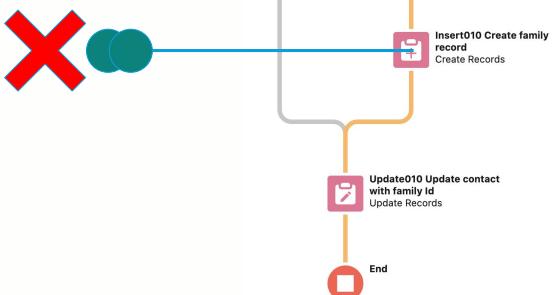
- Creates duplicate records!

DML Operation



Pérez

Pérez





Use Apex In Flows

Can We Use Flows or Do We Need Apex? (Both)



	Low-Code → Pro-Code			Apex Trigger
			+ Apex Class	
Same-Record Field Updates	Available	Not Ideal	Not Ideal	Available
Related Records (CRUD)	Not Available	Available	Available	Available
Asynchronous Processing	Not Available	Available	Available	Available
Complex List Processing (Map, Set)	Not Available	Not Ideal	Available	Available
High-Performance Batch Processing	Not Ideal	Not Ideal	Not Ideal	Available
Custom Validation Errors	Not Available	Not Available	Not Available	Available

<https://architect.salesforce.com/decision-guides/trigger-automation>

@InvocableMethod



Only one per class

Class Access

```
1 public without sharing class RT_LinkFamily {  
2     @InvocableMethod(label='Link Families' description='...'  
3         category='Contact' [Other modifiers])  
4     public static List<Contact> linkFamily(List<Contact> contacts) {  
5         ...  
6     }  
7 }
```

You must Bulkify this!

Must be a list with
same size (or null)

Only one parameter
And it must be a list

Exercise 2

Populate family lookup field (Fixed)



@InvocableVariable And @AuraEnabled



```
1 public with sharing class RT_Demo {  
2     @InvocableMethod(label='Sample' description='...' category='Contact')  
3     public static List<Results> execute(List<Request> requests) {  
4         List<Results> responseWrapper = new List<Results>();  
5         Results response = new Results();  
6         response.outputContact = new RT_DemoAE();  
7         response.outputContact.myContact = requests[0].inputContacts[0];  
8         responseWrapper.add(response);  
9         return responseWrapper;  
10    }  
11    public class Results {  
12        @InvocableVariable(label='Output Contact')  
13        public RT_DemoAE outputContact;  
14    }  
15    public class Request {  
16        @InvocableVariable(label='Input Contact' required='true')  
17        public List<Contact> inputContacts;  
18    }  
19 }
```

User types

First interview

First Record

Inner Classes

@InvocableVariable

Data types for flows
Must be a different file (outer class)

```
1  public class RT_DemoAE {  
2      @AuraEnabled  
3      public Contact myContact { get; set; }  
4  }
```

Exercise 3

@invocableMethods with Complex data





Screen Flows

Three use cases

- LWC (Flow)
- Flow (LWC)
- Custom Property Editors



LWC with Flows



LWC

Flow

Why?

- Higher level of control where the flow is available
- Handle flow events (finish event, ...)

LWCs with Flows



```
1 handleStatusChange(event) {  
2   if (event.detail.status === 'FINISHED') {  
3     let outputVariables = event.detail.outputVariables;  
4     // set behavior after a finished flow interview  
5   }  
6 }
```

Handle events

Receive output variables

Select flow to execute at runtime

```
1 <template>  
2   <lightning-flow flow-api-name={flowName} flow-finish-behavior={behavior}  
3     flow-interview-id={pausedFlowInterviewId} onstatuschange={handleStatusChange}  
4     flow-input-variables={inputVariables}></lightning-flow>  
5 </template>
```

Customize the finish behavior

Start or resume a flow

Custom handling events

Input variables

LWC

Flow

Flows with LWC



Higher control on the form displayed

Flow

LWC

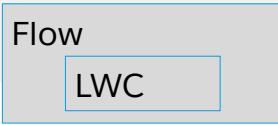
Why?

- Advanced UI (pixel perfect, 4+ columns, tabs)
- Events (onblur, onchange, navigation)
- Form-level interactions (conditional visibility, requiredness, formatting)

Flows with LWC



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
3   <apiVersion>57.0</apiVersion>
4   <isExposed>true</isExposed>
5   <masterLabel>Best Component Ever</masterLabel>
6   <targets>
7     <target>lightning__FlowScreen</target>
8   </targets>
9   <targetConfigs>
10    <targetConfig targets="lightning__FlowScreen">
11      <property name="startDate" label="Start Date" type="Date" role="inputOnly"/>
12      <property name="name" label="Account Name" type="String" role="outputOnly"/>
13      <property name="account" label="Account" type="@salesforce/schema/Account"/>
14      <property name="data" label="Custom Data" type="apex://CustomClass"/>
15    </targetConfig>
16  </targetConfigs>
17 </LightningComponentBundle>
```



Available for flow

Input/output only

Complex data types

Events from LWC to Flow



```
1 import { LightningElement, api, track} from 'lwc';
2 import { FlowAttributeChangeEvent, FlowNavigationNextEvent } from 'lightning/flowSupport';
3
4 export default class LwcInFlow extends LightningElement {
5   @track _someData;
6   @api availableActions = [];
7   @api get someData() { return this._someData; }
8   set someData(value) { this._someData = value; }
9 ...
10 demo() {
11   this.dispatchEvent(new FlowAttributeChangeEvent('someData', this._someData));
12 ...
13   if (this.availableActions.find((action) => action === 'NEXT'))
14     this.dispatchEvent(new FlowNavigationNextEvent());
15 }
16 }
```

Available actions

Control flow navigation
Back, Next, Pause, Finish

Notify values changed
(Beta Spring '23)

Flow

LWC

Exercise 4

Display contacts grouped by families



LWC As Custom Property Editors



Without Custom Property

Edit Apex Action

Use values from earlier in the flow to set the inputs for the "Get Families" Apex action. To use its outputs later in the flow, store them in variables.

Apex010 Get families with contacts (Apex010)

Calls Apex to get related records, and returns the maximum number of records desired

Set Input Values

* Families input
({!Get010})

Count families
150

Max contacts per family
10

> Advanced

Cancel Done

With Custom Property

Edit Apex Action

Apex010 Get families with contacts (Apex010)

Calls Apex to get related records, and returns the maximum number of records desired

Data

families

Get010 Find families

✓ Get010 Find families
Queries all the family records

Max Record Counts

Families
1-200

Contacts
1-20

> Advanced

Cancel Done

LWC As Custom Property Editors



```
1 import { api, LightningElement } from "lwc";
2 export default class CustomConfiguration extends LightningElement {
3   @api inputVariables; Current values to be edited
4   @api builderContext; Flow definition
5   @api
6   validate() {
7     let errorString Validate inputs on "Done"ield. Select a value";
8     this.refs.families.setCustomValidity(errorString);
9     this.refs.families.reportValidity();
10    return [{ key: "families", errorString }];
11  }
12  Array(3)
13  ▶ 0: {name: "countFamilies", value: "150", valueDataType: "Number"}
14  ▶ 1: {name: "families", value: "Get010", valueDataType: "reference"}
15  ▶ 2: {name: "countContacts", value: "10", valueDataType: "Number"}
16  ▶ length: 3
17  ▶ [[Prototype]]: Array(0)
18 }
```

Flow only shows count - Display errors on screen

Return array of errors

Report values changed

Use `@api` properties to capture data from flow builder
Use `events` to report changes to flow builder

What is builderContext?



Actions

Describes flow allowing reflection (introspection)

Get Records

Screens

Variables

```
* Object
  ▶ actionCalls: Array(1)
    ▶ 0: {name: 'Apex010', description: 'Calls Apex to get related records, and re...', length: 1
      ▶ [[Prototype]]: Array(0)
  ▶ apexPluginCalls: []
  ▶ assignments: []
  ▶ choices: []
  ▶ collectionProcessors: []
  ▶ constants: []
  ▶ decisions: []
  ▶ dynamicChoiceSets: []
  ▶ formulas: []
  ▶ loops: []
  ▶ recordCreates: []
  ▶ recordDeletes: []
  ▶ recordLookups: Array(1)
    ▶ 0: {name: 'Get010', description: 'Queries all the family records', label: 'Get...', length: 1
      ▶ [[Prototype]]: Array(0)
  ▶ recordRollbacks: []
  ▶ recordUpdates: []
  ▶ screens: Array(1)
    ▶ 0: {name: 'Screen010', description: '', label: 'Screen010', locationX: 176, length: 1
      ▶ [[Prototype]]: Array(0)
  ▶ stages: []
  ▶ start: {name: undefined, description: undefined, label: undefined, locationX: 176, length: 1
    ▶ [[Prototype]]: Object()
  ▶ textTemplates: []
  ▶ variables: Array(1)
    ▶ 0: {name: 'familyTypes', description: '', apexClass: 'UI_FamilyTypeAE', data...', length: 1
      ▶ [[Prototype]]: Array(0)
  ▶ waits: []
    ▶ [[Prototype]]: Object()
```



Exercise 5

LWC As Custom Property Editors



Thank you

