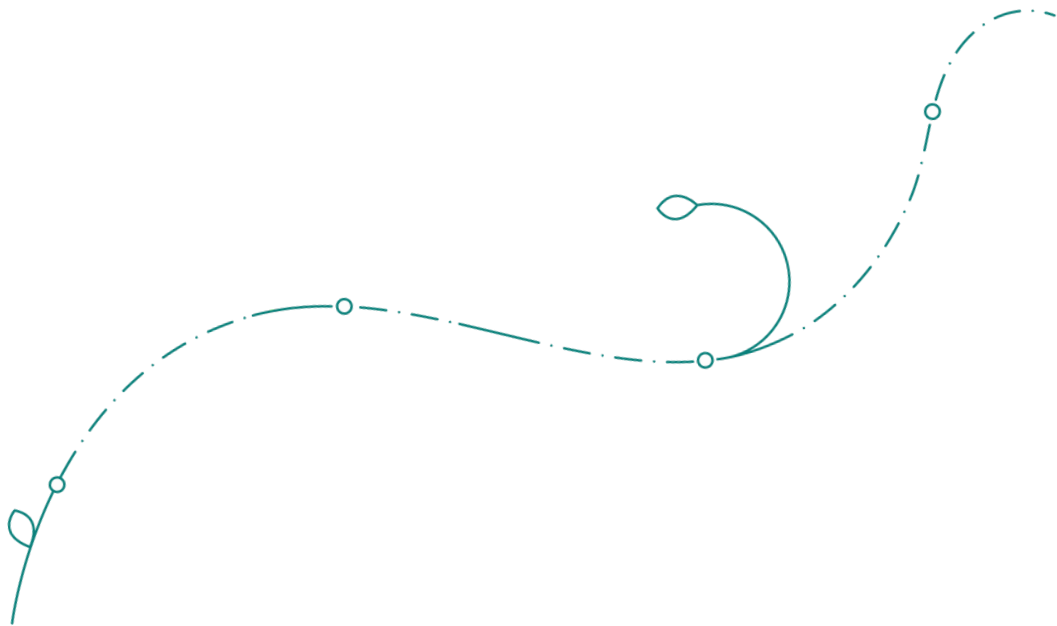


# Flow + Apex + LWC

A Perfect Combination!

EXERCISE WORKBOOK



## Table of Contents

Exercise 1: Populate family lookup field	2
Exercise 2: Populate family lookup field (Fixed)	6
Exercise 3: @invocableMethods with Complex data structures	10
Exercise 4: Display contacts grouped by families	16
Exercise 5: LWC As Custom Property Editors	22

### Exercise 1: Populate family lookup field

**Goal:** Attempt to populate the Family\_\_c lookup up field on the Contact record based on the last name

#### Instructions:

1. First, let's see if this works
  - Open the **EX01 - Link family** flow
  - Click **Debug**
  - Select **Skip start condition requirements**
  - Select **Updated**
  - Type: **Perez**
  - Select any contact
  - Change Last Name to **Smith**
  - Click **Run**

- a. Which path did the flow go through?

- b. Are the new records permanently stored in the database?

- c. Why are the records not permanently in the database?

- Click **Edit Flow** and **Activate** the flow

## EXERCISE GUIDE

Session Name



### 2. Review the **Start** element

- Go back to the **EX01 - Link family** flow
- Review the **Start** element

a. What type of flow is this?

b. What object is this record-trigger flow firing on?

### 3. Review the **Get010** element

a. What is the equivalent SOQL query?

SELECT \_\_\_\_\_  
FROM \_\_\_\_\_  
WHERE \_\_\_\_\_  
ORDER BY \_\_\_\_\_  
LIMIT \_\_\_\_\_

b. Where do the results get stored?

### 4. Review the **IF020** element

a. What is this condition validating?

5. Review the **Insert030** element

- a. Which record does it create?

- b. Where do the results get stored?

6. Review the **Update40** element

- a. Which record does it update?

- b. Which field is being updated?

7. Does it **really** work?

- Open **HOW/main/default/classes/EX01\_LinkFamily\_Test.cls**

- a. What does the **testSetup()** method do in this file?

- b. How many contact records should be there **during the test execution**?

- c. How many Family\_\_c records should be there **during the test execution**?

## EXERCISE GUIDE

Session Name



- Other methods will create more Contacts with different last name scenarios
- In line #1.5 click **Run All Tests**
- d. Did all the tests succeed?

- In line #25, remove the **comment**
- **Save and deploy**
- In line #23.5 click **Run Test**
- e. Did the tests succeed?

- f. What error did we get?

### 8. Deactivate flow

- Go back to the flow and **deactivate** it!

### Exercise 2: Populate family lookup field (Fixed)

**Goal:** Use an @invocable Apex method to fix the issue

#### Instructions:

1. First, let's see if this works
  - Open the **EX02 - Link family (Apex)** flow
  - Activate this flow
  - Open **HOW/main/default/classes/EX01\_LinkFamily\_Test.cls**
  - In line #1.5 click **Run All Tests**
  - a. Did all the tests succeed?
  - b. Did we run the test on line 25? (remember it was failing before)
  - c. Was the test on line 25 successful?
2. Review the **Apex010** element
  - Go back to the **EX02 - Link family (Apex)** flow
  - Review the **Apex010** element
  - a. What is the name of the global variable we are sending to Apex?
  - b. How many Contact records does **\$Record** hold?

- Expand the **Advanced** section
  - c. What is the name of the variable receiving the results from the Apex call?
  - d. What is the data type of **contactWithFamily**?
  - e. How many Contact records can **contactWithFamily** hold?
3. Review the **Update020** element
- a. What is the variable name that holds the information we are updating?
  - b. How many Contact records can **contactWithFamily** hold?
4. Review the **EX02\_LinkFamily.cls** Apex class
- Open **HOW/main/default/classes/EX02\_LinkFamily.cls**
  - a. In line #2, which annotation are we using?
  - b. How many methods annotated with **@InvocableMethod** are in this class?



- c. In line #3, how many parameters take the method?

- d. What's the data type of the parameter **contacts**?

- e. How many Contact records can this variable hold?

- f. What's the return type?

- g. How many Contact records could be returned?

- h. From the flow, we are passing one record, but we are receiving many! **Why?**

- i. In line #25, which records are we inserting in that DML operation?

- j. Which line is doing the DML of the contacts? (*tricky question*)

5. Review the permission set

- Open **HOW/main/default/permissionsets/HOW.permissionset-meta.xml**
- a. In line #8, which class are we enabling?

### Exercise 3: @invocableMethods with Complex data structures

**Goal:** Review how complex data structures work with @invocableMethods

**Instructions:**

1. First, let's see if this works

- Open the **EX03 - DataStructures** flow
- Click **View Tests**
- In the list, click on the dropdown and select **Edit**
- Click on the **Set Assertions** link
- a. On the first assertion, which property of results contains the LastName being compared with "Pérez"?

- Click **cancel**
- In the list, click on the drop-down and select **Run Test and View Details**
- b. Did both assertions pass?

2. Review the **Set010** element

- Go back to the **EX03\_DataStructures** flow
- Review the **Set010** element
- a. How many Contact records can **\$Record\_\_Prior** hold? (Assume we are updating)

- b. How many Contact records can **\$Record** hold?

- c. What is the data type of the variable **contacts**?

- d. How many Contact records can **contacts** hold?

- e. What are we doing in this assignment element?

- f. How many pairs of Contact records can **contacts** hold?

- g. We don't have a loop! How are we adding multiple pairs? (tricky question)

3. Review the **Apex020** element

- a. How many input values are we sending?

- b. Is the first value (**Input Contacts**) required?

c. What is the variable name we are sending?

d. Is the second value (**The answer**) required?

e. What value are we passing in the second value (**The answer**) ?

f. Is that the right answer? (Geeky joke)

- Expand the **Advanced** section

g. What is the variable name receiving the result from the Apex call?

h. What is the data type of **results**?

- i. Does this variable accept one or many elements?

- j. Why just one?

4. Review the **EX03\_DataStructures.cls** Apex class

- Open **HOW/main/default/classes/EX03\_DataStructures.cls**

- a. In line #2, which annotation are we using?

- b. What's the data type of the parameter **requests**?

- c. What determines the number of elements in **requests** parameter?

- d. Where (which line number) are we defining the **Request** class?

- e. How many class variables are defined in the **Request** class?

f. Which annotation are we using for them?

g. In line #7, what do the **requests[0]** mean?

h. What do the **requests[0].inputContacts[0]** mean?

i. In line #3, what's the return type for the **@invocableMethod**?

j. What determines the number of elements returned?

k. Where (which line number) are we defining the **Result** class?

l. How many class variables are defined in the **Result** class?

m. Which annotation are we using for **outputData**?

n. What's the data type of **outputData**?

5. Review the **EX03\_DataStructuresAuraEnabled.cls** Apex class

- Open **HOW/main/default/classes/EX03\_DataStructuresAuraEnabled.cls**

a. How many properties are in the **EX03\_DataStructuresAuraEnabled** class?

b. Which annotation are we using for them?

- In question 1a, We saw the assertions used results.**current**.

- Thanks, @AuraEnabled!



### Exercise 4: Display contacts grouped by families

**Goal:** Using a LWC display the Family\_\_c records grouped by last name and show their related Contact records

#### Instructions:

1. First, let's see if this works

- Navigate to the Family application
- Open the home tab
- Click on the **Finish** button

a. Which error did we get?

- Find the **Pérez** family

b. How many Contact records are associated with this last name?

- Expand the **Pérez** family
- Click on **Andrés**

c. Did we navigate to the **Andrés Pérez** Contact record?

2. Review the **Get010** element

- Open the **EX04 - TreeGrid** flow
- Review the **Get010** element
- a. What is the equivalent SOQL query?

SELECT \_\_\_\_\_  
FROM \_\_\_\_\_  
WHERE \_\_\_\_\_  
ORDER BY \_\_\_\_\_  
LIMIT \_\_\_\_\_

- b. Where do the results get stored?

3. Review the **Apex020** element

- a. How many values are we sending?

- b. What is **Families Input**?

- c. What else are we sending?

- d. Where do the results get stored?

4. Review the **EX04\_QueryFamilies.cls** Apex class

- Open **HOW/main/default/classes/EX04\_QueryFamilies.cls**

a. In line #12, we have an assertion. What does it validate?

b. Why can we make sure this assertion always passes?

c. In line #15 which Family\_\_c records do we query?

d. Is there a maximum number of Family\_\_c records we retrieve?

e. In line #15 which Contact records do we query?

f. Is there a maximum number of Contact records we retrieve per Family\_\_c?

g. In line #3, what is the return type of the method?

- h. Which line number defines the **FamilyResponse** class?

- i. What is the name of the class variable defined in the **FamilyResponse** class?

5. Review the **Screen030** element

- Go back to the **EX04 - TreeGrid** flow and review the **Screen030** element

- a. How many custom components do we have in this flow?

- b. What is the name of the custom component we are using?

- Select the **Family Tree Grid** component

- c. What is the data we are using in **Family Types**?

- d. Why can we use **Apex020.response**?

6. Review the **ex04TreeGrid** LWC component

- Find the **HOW/main/default/lwc/ex04TreeGrid** folder
- Open the **ex04TreeGrid.html** file

a. Which standard component are we using?

- Open the **ex04TreeGrid.js-meta.xml** file

b. In line #5, what is the **masterLabel**?

c. In line #7, what is the **target**?

d. In line #11, what is the value of the property's **label** attribute?

e. In line #11, what is the value of the property's **type** attribute?

f. In line #11, what is the value of the property's **name** attribute?

- Open the **ex04TreeGrid.js** file

g. Are we using a **@wire**?

h. How do we get the data without a @wire?

i. In line #57, what's the error message?

j. In line #67, what are we doing?

## Exercise 5: LWC As Custom Property Editors

**Goal:** Use LWCs to configure the admin view of the flows.

### Instructions:

1. First, let's see if this works

- Open the **EX05 - Custom Properties** flow
- New path has been added, the old path is the same as Exercise 4
- Open the **Apex020 (New)** element
- a. Which standard LWC do we use to input **Families Data Provider**?

b. What are the options available?

- Select **None** and click **Done**

c. What error did you get?

- Switch back to Get010

d. Which standard LWC do we use to input **Max Families**?

e. Which standard LWC do we use to input **Max Contacts Per Family**?

## EXERCISE GUIDE

Session Name



- Open the **Apex020 (Old)** element

f. Are the input widgets different?

- Open the **Screen030 (New)** element

- Click on **Family Tree Grid (EX05)**

g. Which standard LWC do we use to input **Data**?

h. How many options are available?

- Select **None** and click **Done**

i. Did you get an error?

- Open the **Screen030 (New)** element

- Switch back to **Apex020 (New) Get families with contacts**

- Click **Done**

- Open the **Screen030 (Old)** element

- Click on **Family Tree Grid**

j. Is the input widget for **Family Types** different?



2. Review the **EX05\_QueryFamilies.cls** Apex class

- Open **HOW/main/default/classes/EX05\_QueryFamilies.cls**
- This class is identical to the previous one except for one difference...
- a. In line #6, which attribute of the **@invocableMethod** are we configuring?

- b. In line #6, what's the value for **configurationEditor**?

3. Review the **ex05ApexConfiguration** LWC component

- Find the **HOW/main/default/lwc/ex05ApexConfiguration** folder
- Open the **ex05ApexConfiguration.html** file
- a. In line #5, which LWC do we use to input **Families Data Provider**?

- b. In line #20, which LWC do we use to input **Max Families**?

- c. In line #31, which LWC do we use to input **Max Contacts Per Family**?

- Open the **ex05ApexConfiguration.js** file

- d. In line #26, which setter do you find?

- e. In line #45, which setter do you find?

- f. In line #49, how do we find the list of **Get** elements in the flow?

- g. In line #65, what is the error message?

- h. In line #99, how does LWC notify the flow that we selected a different value?

4. Review the **ex05TreeGrid** LWC component

- Find the **HOW/main/default/lwc/ex05TreeGrid** folder
- Open the **ex05TreeGrid.js-meta.xml** file
- a. In line #10, which **configurationEditor** are we specifying?

- Find the **HOW/main/default/lwc/ex05LwcConfiguration** folder
- Open the **ex05LwcConfiguration.html** file
- b. In line #5, which LWC do we use to input **Data**?

- Open the `ex05LwcConfiguration.js` file

- c. In line #16, which setter do you find?

- d. In line #34, which setter do you find?

- e. In line #39, how do we find the list of **Apex Action** elements in the flow?

- f. In line #56, how does LWC notify the flow that we selected a different value?