

# DF24 - Thursday 2024.09.19

## DevFlow (Flows + Apex + LWC)

### Exercise 1: Populate family lookup field

1. Which path did **IF020** go through?

- ☒ A. Family must be created
- ☐ B. Family Found

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

- ☐ A. Account
- ☒ B. Contact
- ☐ C. Family\_\_c

3. Which DML operations are firing this record trigger flow?

- ☒ A. Insert
- ☒ B. Update
- ☐ C. Delete
- ☐ D. Undelete

4. Which criteria must be met for this record-triggered flow to execute? (Entry condition)

- ☐ A. Every Contact
- ☒ B. Orphan contacts with last name
- ☒ C. Contacts with last name but different from the associated parent record.

5. What type of flow is this?

- ☐ A. French fries flow
- ☒ B. Pirate flow
- ☐ C. Screen flow

6. Which object is this querying?

- ☐ A. Account
- ☐ B. Contact
- ☒ C. Family\_\_c
- ☐ D. Person\_\_c

7. We are sorting the results

- ☐ A. True
- ☒ B. False

8. A similar SOQL is being executed in **Get010**

```
SELECT FIELDS(All)
FROM Family__c
WHERE LastName__c = :$Record.LastName
LIMIT 1
```

- ☒ A. True
- ☐ B. False

9. What is the API name of the variable that holds the results? (Hint: Is the same name as the flow element)

- ☒ A. Get010
- ☐ B. Families
- ☐ C. {!\$SOQLResult}

10. What is this condition validating?

- ☐ A. If the contact has a last name
- ☒ B. If there is a family with this last name

11. Which record does it **create**?

- ☐ A. Account
- ☐ B. Contact
- ☒ C. Family\_\_c (The missing family)

12. Which record does it **update**?

- ☐ A. Account
- ☒ B. Contact (The contact record that triggered the flow)
- ☐ C. Family\_\_c

13. All tests successfully passed

- ☒ A. True
- ☐ B. False

14. All tests successfully passed

- ☐ A. True
- ☒ B. False

15. What error did we get?

- ☐ A. Governor limit exception
- ☐ B. Required field missing
- ☒ C. Duplicate values
- ☐ D. Other

Exercise 2: Populate family lookup field (Fixed)

1. What type of flow is this?
  - ☐ A. French fries flow
  - ☒ B. Pirate flow
  - ☐ C. Screen flow
2. All tests successfully passed
  - ☒ A. True
  - ☐ B. False
3. What is the name of the variable we are sending to Apex?
  - ☒ A. \$Record
  - ☐ B. \$Record\_\_prior
  - ☐ C. Family\_\_c
4. How many Contact records does **\$Record** hold?
  - ☒ A. One
  - ☐ B. Many
5. What is the name of the variable receiving the results from the Apex call?
  - ☒ A. contactWithFamily
  - ☐ B. \$Record\_\_prior
  - ☐ C. Family\_\_c
6. What's the **contactWithFamily** data type?
  - ☒ A. Contact
  - ☐ B. List<Contact>
7. What is the variable name that holds the information we are updating?
  - ☒ A. contactWithFamily
  - ☐ B. \$Record\_\_prior
  - ☐ C. \$Record
8. In line #5, which annotation are we using?
  - ☐ A. @IsTest
  - ☐ B. @Future
  - ☒ C. @InvocableMethod
  - ☐ D. @InvocableVariable
9. How many methods are annotated with **@InvocableMethod**?
  - ☒ A. 1
  - ☐ B. 2
10. In line #6, how many parameters take the method?
  - ☒ A. 1
  - ☐ B. 2

11. What's the data type of the parameter contacts?

- ☐ A. Contact
- ☒ B. List<Contact>

12. How many Contact records can this variable hold?

- ☐ A. One
- ☒ B. Many

13. What's the method return type?

- ☐ A. Contact
- ☒ B. List<Contact>

14. How many Contact records could be returned?

- ☐ A. One
- ☒ B. Many

15. From the flow, we are passing and receiving only one record, but here we are receiving and returning many! Why?

- ☒ A. Automatic Bulkification
- ☒ B. Flow Interview vs. Flow Transaction
- ☐ C. Contacts have last names
- ☐ D. Family record have last names

16. In line #44, which DML operation are we performing?

- ☒ A. Insert missing families
- ☐ B. Update all families
- ☐ C. Delete old contacts

17. Which line is doing the DML of the contacts? (tricky)

- ☐ A. 17
- ☐ B. 44
- ☒ C. None, the DML is performed by the flow

18. In line #8, which class are we enabling?

- ☐ A. EX01\_LinkFamily\_Test
- ☒ B. EX02\_LinkFamily
- ☐ C. EX03\_QueryFamilies

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

1. What type of flow is this?

- ☐ A. French fries flow
- ☐ B. Pirate flow
- ☒ C. Screen flow

2. What value is being displayed for **The answer**?

☒ A. 42

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

☒ A. Yes

4. Which object is this querying?

☒ A. Family\_\_c

☒ B. Contact

☒ C. Account

5. What is the API name of the variable that holds the results? (Hint: Is the same name as the flow element)

☒ A. Get010

☒ B. Results

☒ C. {!\$SOQLResult}

6. How many input values are we sending?

☒ A. 0

☒ B. 2

☒ C. 4

7. What value are we sending for **Family Inputs**?

☒ A. Only one family

☒ B. {!Get010} (All the families queried)

8. Since **Max Families** value is not required, how can we indicate we are not passing a value?

☒ A. Value is required

☒ B. Leave the textbox empty

☒ C. Just toggle the switch

9. What is the variable name receiving the result from Apex?

☒ A. response

☒ B. Get010

☒ C. Apex020

10. What is the data type of **response**?

☒ A. List<EX03AE\_Response>

☒ B. EX03AE\_Response

☒ C. List<Family\_\_c>

11. What is the expression displayed in the display text element?

☒ A. {!response.theQuestion}

☒ B. {!response.theAnswer}

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

- ☐ A. @IsTest
- ☐ B. @Future
- ☒ C. @InvocableMethod
- ☐ D. @InvocableVariable

13. What's the data type of the parameter **familyRequests**?

- ☒ A. List<FamilyRequest>
- ☐ B. FamilyRequest

14. In line #12, what does the **familyRequests[0]** mean?

- ☐ A. The first Family\_\_c record
- ☒ B. The first Flow Interview
- ☐ C. The first Flow Transaction

15. In line #11, we have an assertion. Will this ever fail? (tricky)

- ☐ A. Yes, if we have zero Flow Interviews
- ☐ B. Yes, pirate flows create Flow Interviews for each record DML
- ☒ C. No, this is a screen flow (one and only one Flow Interview)

16. Where (which line number) are we defining the FamilyRequest class?

- ☐ A. 3
- ☒ B. 26
- ☐ C. 33

17. How many variables are defined in the **FamilyRequest** class?

- ☐ A. 1
- ☒ B. 2
- ☐ C. 3

18. Which annotation are we using for them?

- ☐ A. @IsTest
- ☐ B. @Future
- ☐ C. @InvocableMethod
- ☒ D. @InvocableVariable

19. What's the **label** for the first variable? (Line 27)

- ☒ A. Families input
- ☐ B. Families output
- ☐ C. maxFamilies

20. What's the **families** data type? (Line 28)

- ☐ A. Family\_\_c
- ☒ B. List<Family\_\_c>
- ☐ C. Integer

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

- ☐ A. Family\_\_c
- ☐ B. List<Family\_\_c>
- ☒ C. List<FamilyResponse>

22. How many elements, and in which order, should that list contain?

- ☒ A. Must correspond (number, order) to the elements in the parameter
- ☐ B. The number and order are not relevant
- ☐ C. Any number and order is fine

23. Which line number is the **FamilyResponse** defined?

- ☐ A. 3
- ☐ B. 26
- ☒ C. 33

24. Which annotation are we using for **response**?

- ☐ A. @IsTest
- ☐ B. @Future
- ☐ C. @InvocableMethod
- ☒ D. @InvocableVariable

25. What's the data type of **response**?

- ☐ A. FamilyResponse
- ☒ B. EX03AE\_Response
- ☐ C. EX03\_QueryFamilies

26. How many properties are in the **EX03AE\_Response** class?

- ☐ A. 1
- ☒ B. 2
- ☐ C. 3

27. Which annotation are we using for them?

- ☐ A. @IsTest
- ☒ B. @AuraEnabled
- ☐ C. @InvocableMethod
- ☐ D. @InvocableVariable

#### Exercise 4: Display contacts grouped by families

1. What happened?

- ☒ A. Error: "You can't go past this screen!"
- ☐ B. Navigated to the next screen
- ☐ C. Flow completed successfully

2. What happened?

- ☐ A. Error: "You can't navigate!"
- ☒ B. Navigated to the Andrés Pérez contact record
- ☐ C. Nothing!

3. What type of flow is this?

- ☐ A. French fries flow
- ☐ B. Pirate flow
- ☒ C. Screen flow

4. What is the name of the custom component?

- ☐ A. Display Text
- ☒ B. Family Tree Grid
- ☐ C. There is none!

5. What is the label for the textbox containing the expression **{!response}**?

- ☐ A. API Name
- ☒ B. Family Data
- ☐ C. Family Types

6. Which standard component are we using?

- ☒ A. <lightning-tree-grid />
- ☐ B. <lightning-button />
- ☐ C. <lightning-card />

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

- ☐ A. ex04TreeGrid.js
- ☐ B. Undefined
- ☒ C. Family Tree Grid

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

- ☒ A. lightning\_\_FlowScreen
- ☐ B. lightning\_\_HomePage
- ☐ C. lightning\_\_RecordPage

9. In line #13, what does **name="familyData"** represent?

- ☒ A. The JavaScript variable
- ☐ B. The label for the property in the screen component
- ☐ C. The expected data type that flow should pass

10. In line #13, what does **label="Family Data"** represent?

- ☐ A. The JavaScript variable
- ☒ B. The label for the property in the screen component
- ☐ C. The expected data type that flow should pass



11. In line #13, what does **type="apex://EX03AE\_Response"** represent?

- ☐ A. The JavaScript variable
- ☐ B. The label for the property in the screen component
- ☒ C. The expected data type that flow should pass

12. How do we get the families' data?

- ☐ A. @wire (data from Apex)
- ☒ B. @api familyData (data from the flow)
- ☐ C. Imperative Apex (data from Apex)

13. In line #59, what's the error message?

- ☒ A. You can't go past this screen!
- ☐ B. You can't view contacts

14. In line #77, what are we doing?

- ☒ A. Navigating to the Contact record
- ☐ B. Displaying an error message