We love feedback!

Salesforce Hands-on Training

Enhance Flows with Apex and Lightning Web Components

Flow + Apex + LWC = A Perfect Combination!

EXERCISE GUIDE

Learn more at: http://sfdc.co/successplanresources

Table of Contents

.

Exercise 0: Set up your environment for this session

Register with the session's online site

1. Using Chrome, navigate to **https://sfdc.co/devFlow**.
2. Select **Continue**.
3. Enter your **First Name**.
4. Enter the **Computer Number**.
5. Select Select**Register**.
6. Go to the session page and Select the split screen icon. ⤢
    - Left: **Session page**.
    - Right: **Working area** (Org + Code Builder).

Log in to the specially-provided Salesforce trial org

7. On the right screen navigate to https://login.salesforce.com/.
8. Enter the username and password login information provided to you by your instructor.
9. Select **Log in to Salesforce**.

Enable Salesforce Code Builder in your Salesforce trial org

10. In the upper right, open **Setup**.
11. In the upper-left Quick Find box, navigate to **Development | Code Builder**.
    Note: If you see an alert box open, select the upper-right corner X to close it.
12. Set the toggle switch to **Enabled**.
13. In the **Product Terms** dialog box, select **Accept**.

Launch Code Builder

14. From the App Launcher in the upper left, open **Apps | Code Builder**.
15. Select **Launch**.
16. Wait for Code Builder to load (it may take a few minutes).

.

Exercise 1: Analyze and test a Salesforce flow

Debug a pre-built Salesforce flow

1. Return to the Salesforce page and in the upper right open **Setup**.
2. In the upper-left Quick Find box, navigate to **Process Automation | Flows**
3. Scroll through the flows and Select the **EX01 - Link family** link.
4. In Flow Builder, in the upper right, select **Debug**.
5. In **Run the Flow As If the Record is:** select the **Updated** radio button.
6. In **Debug Options**, check the **Skip start condition requirements** checkbox**.**
7. In the **Contact** search box, type **Perez**.
8. Select any contact.
9. Scroll to the bottom and change the required **Last Name** field to **Smith**.
10. Select **Run.**
    - Answer QUESTION #1.

Activate the flow

11. In the top, select **Edit Flow** to go back to the **EX01 - Link family** flow.
12. In the top right, select **Activate**.

Analyze how the flow's Start element triggers the flow

13. Select the **Start** element and Select **Edit**; the Start editor should open on the right.
    - Answer QUESTIONS #2 through #5.

Analyze what the flow's Get010 element does

14. In the flow on the left, select and edit the **Get010** element.
    - Answer QUESTIONS #6 through #9.

Analyze what the rest of the flow does

15. In the flow, select and edit the **IF020** element
    - Answer QUESTION #10.
16. Select and edit the **Insert030** element.
    - Answer QUESTION #11.
17. Select and edit the **Update40** element.
    - Answer QUESTION #12.

Authorize your Salesforce trial org with Code Builder

18. Return to Code Builder and from the main menu select **View | Command Palette**.
19. Use typeahead to select the command `> SFDX: Authorize an Org`

.

20. Select **Project Default**.
21. Enter the alias `df24apexflows`.
22. In **Enter Code**, select **Connect**; a Salesforce login tab should open.
23. Log in with the same credentials that you previously used to log in to the org.
24. In **Allow Access?**, select **Allow**.
25. In **Your Connected**, select **Continue**.

Create a Code Builder project with the deployed flows

26. Return to Code Builder.
27. In the Command Palette, type > then use typeahead to select `SFDX: Create Project with Manifest`.
28. Select **Standard Project template**.
29. Enter the name: `df24apexflows`.
30. Change the path to `/home/codebuilder/`.
31. In the lower left Select **No Default Org Set**.
32. In the top Command Palette, select **df24apexflows**.
33. From the main menu, select **Terminal | New Terminal**.
34. Execute the command: `sf retrieve metadata --manifest manifest/package.xml`
    Note: You can copy/paste these commands from a file in the github repository at http://tinyurl.com/df24apexflows.
35. Execute the command: `sf retrieve metadata --metadata PermissionSet:HOW`
    Note: If you have trouble, you can also execute the command: `>git clone` [`http://tinyurl.com/df24apexflows`](http://tinyurl.com/df24apexflows) *to create a new project.*

Test if the exercise flow correctly relates new and updated contact records with correct family records

36. Select force-app/main/default/classes/**EX01_LinkFamily_Test.cls**; the file should open in a new tab.
    Note: The tests cover all scenarios of creating and updating two contact records, and verifies if the referenced family records exist or are created, and the last names match.
37. In line #1.5, just under the outermost @IsTest annotation, select **Run All Tests**.
38. Confirm the results of the tests in the Output pane.
    - Answer QUESTION #13.
39. In the **EX01_LinkFamily_Test.cls** file, remove the comment on line 25 in the **newContacts_WithDuplicate_NewLastNames()** method.

.

40. In the Explorer pane, right-click on **EX01_LinkFamily_Test.cls** and select **SFDX: Deploy This Source to Org**.
41. In line #23.5 Select **Run Test**.
42. Look in the Output pane to see the results of the tests.
    - Answer QUESTIONS #14 through #15.

Deactivate the EX01 - Link family flow

43. Return to the **EX01 - Link family** flow browser tab.
44. Select **Deactivate**.

.

Exercise 2: Analyze how to fix issues and limitations in a flow by using an Apex action element

Open the flow for this exercise in Flow Builder

1. Return to Salesforce Flow Builder.
2. In the upper left, change the drop-down menu to **All Flows**.
3. Open the **EX02 - Link family (Apex)** flow in Flow Builder.
4. Select **Activate**.
   *Note: Be sure you have deactivated the previous **EX01 - Link family** flow.*
   - Answer QUESTION #1.

Run the previous tests to see if the new EX02 - Link Family flow fixes the previous issue when inserting duplicate family records

5. Return to Code Builder.
6. Open force-app/main/default/classes**/EX01_LinkFamily_Test.cls**.
7. In line #25.1, just above the **newContacts_WithDuplicate_NewLastNames()** method, select **Run Test**.
   - Answer QUESTION #2.

   *Note: In line #1.5, you can also select **Run All Tests** to check if all the tests pass with this new flow.*

Analyze how the Apex010 element connects the flow with the Apex code

8. Return to Flow Builder.
9. In the flow, select and edit the **Apex010 Call apex to link contacts** element.
10. In the **Set Input Values for the Selected Action** section, in the **contacts** text box, select **Triggering Contact** to view the variable that is being sent to the Apex method.
    - Answer QUESTIONS #3 through #4.
11. Expand the **Show advanced options** section.
    - Answer QUESTIONS #5 through #6.

Analyze how the Update020 element connects with Apex code

12. Select and edit the **Update020** element.
    - Answer QUESTIONS #7.

Analyze the Apex code used to customize the flow

13. Return to Code Builder.
14. In the Explorer pane, select force-app/main/default/classes**/EX02_LinkFamily.cls**.

.

- Answer QUESTIONS #8 through #17.

Analyze the permission set used with the flow

15. In the Explorer pane, open force-app/main/default/permissionsets**/HOW.permissionset-meta.xml**.
16. In the editor tab, look at the **<classAccesses>** tag.
    - Answer QUESTIONS #18.

.

Exercise 4: Analyze how to code and embed a Lightning Web Component (LWC) in a screen flow

Analyze the behavior of a screen flow that embeds an LWC component

1. Return to the Salesforce org tab.
2. In the upper-left **App Launcher**, navigate to the **Families** application.
3. Select the **Home** tab.
4. In the **Ex04 - TreeGrid** pane, select **Finish**.
   - Answer QUESTION #1.
5. In the **EX04 - TreeGrid** pane, locate the **Perez** family.
6. Expand the **Perez** family.
7. Select **Andrés**.
   - Answer QUESTION #2.

Analyze a screen flow with a screen element that embeds an LWC

8. In App Launcher, navigate to **Flows**.
9. Open the **EX04 - TreeGrid** flow in Flow Builder.
   - Answer QUESTION #3.
10. In the upper-right, if the flow state says **Deactivated**, select **Activate**.
11. Select and edit the **Screen030 Display Apex Output** screen element.
12. In the **Edit Screen** dialog box, in the middle **EX04 - TreeGrid** pane, select the embedded **Family Tree Grid** LWC component.
    - Answer QUESTIONS #4 through #5.
13. Select **Done**.

Analyze how the LWC is built in Code Builder

14. Return to Code Builder.
15. In the left-side Explorer pane, expand the force-app/main/default/lwc**/ex04TreeGrid** folder.
16. Open the **ex04TreeGrid.html** file.
    - Answer QUESTION #5 through #6.
17. Open the **ex04TreeGrid.js-meta.xml** file.
    - Answer QUESTION #7 through #11.
18. Open the **ex04TreeGrid.js** file.
    - Answer QUESTION #12 through #14.

.