Lab 7. Interaction between Power Automate and Power Apps

Lab 7. Interaction between Power Automate and Power Apps

Author: Serge Luca, aka "Doctor Flow"

Learning objective: in many business scenarios, a Power Apps application must call a Power Automate flow because some actions are unavailable on Power Apps, some processes take some time to complete, or because we want a clean boundary between the user interface and the business logic.

Duration: 30 minutes

Scenario: you will create a PowerApps form allowing any user to create a Team. The first part of the lab will call a dummy Flow that you will make. Then, if time permits, you will invoke the Team creation Flow you will create later.

Tasks

1. Create an instant Flow with From Power Apps as the trigger:

Build an instant cloud flow

Flow name

Create a team

Choose how to trigger this flow *

		Manually trigger a flow Flow button for mobile	ⓘ
●		PowerApps PowerApps	ⓘ
		When Power Virtual Agents calls a fl... Power Virtual Agents	ⓘ
		When a flow step is run from a busi... Microsoft Dataverse	ⓘ
		For a selected message (V2) Microsoft Teams	ⓘ
		From the compose box (V2) Microsoft Teams	ⓘ
		When someone responds to an ada...	ⓘ

Triggered manually from any device, easy-to-share instant flows automate tasks so you don't have to repeat yourself.

Examples:
- Get an automatic mobile alert whenever a VIP client emails you
- Save all your email attachments to a folder automatically

Skip   Create   Cancel

2. Remove the Power Apps trigger to use a new version of the trigger.
3. Replace the trigger with the Power Apps v2:
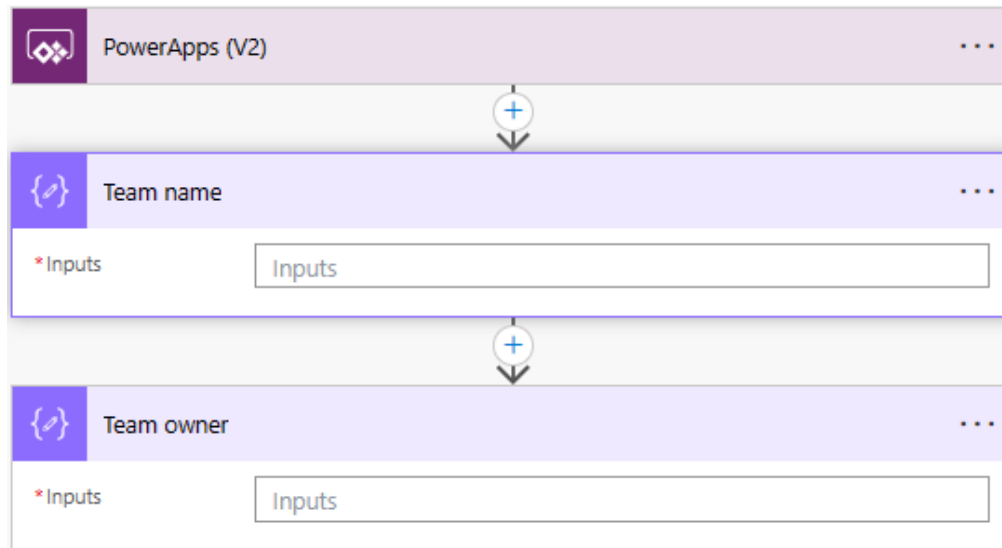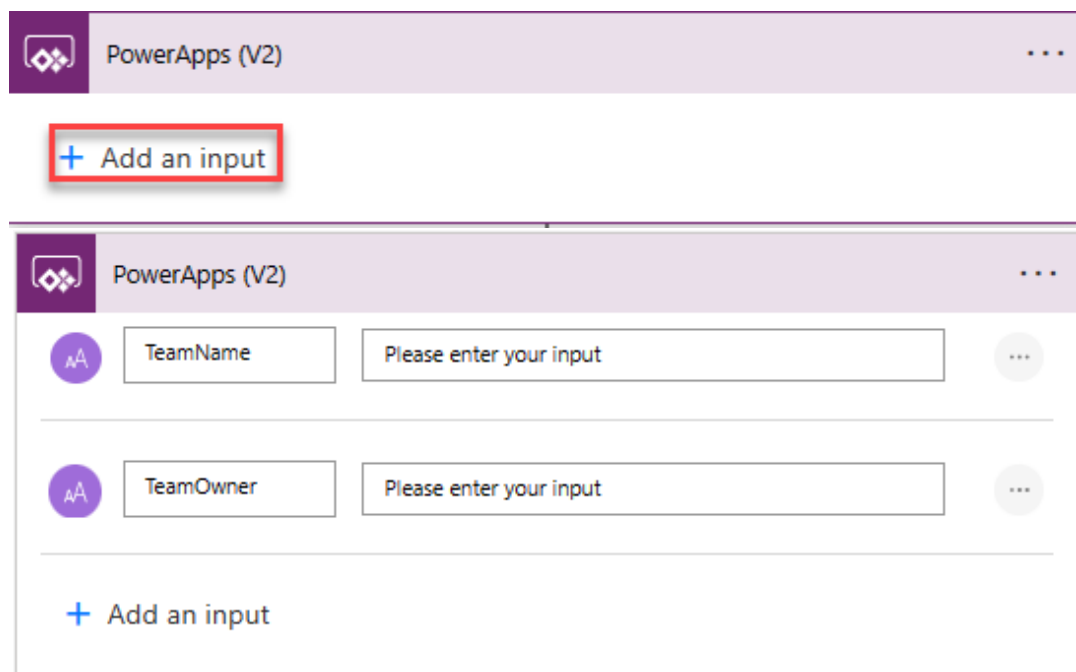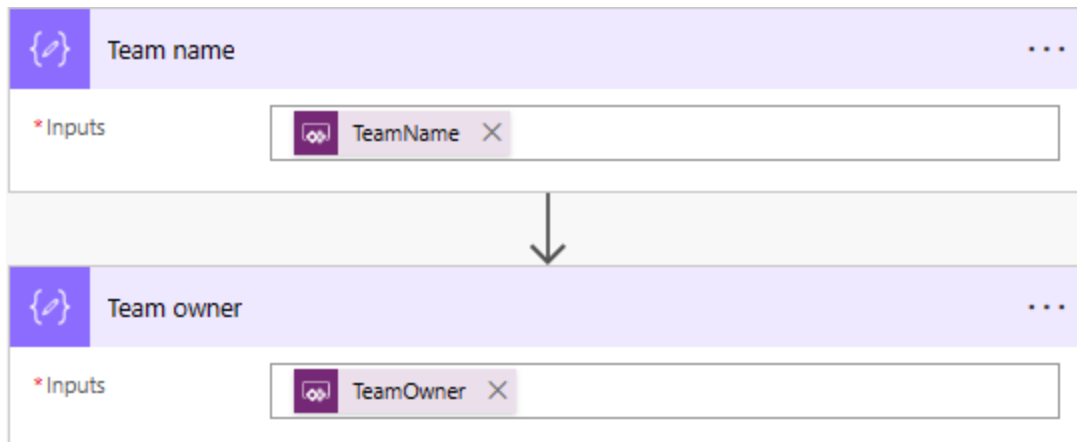

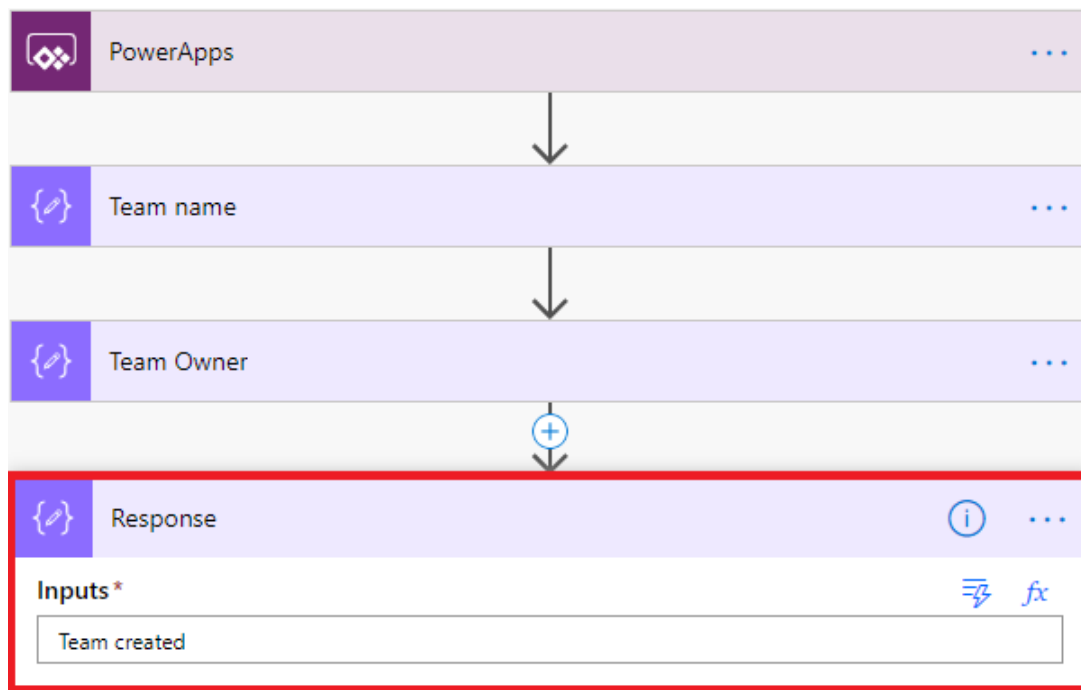4. In this flow, add 2 Compose actions and rename them **Team name** and **Team owner** :

5. Click the trigger to define two parameters of type Text by clicking Add an input, and name them TeamName and TeamOwner
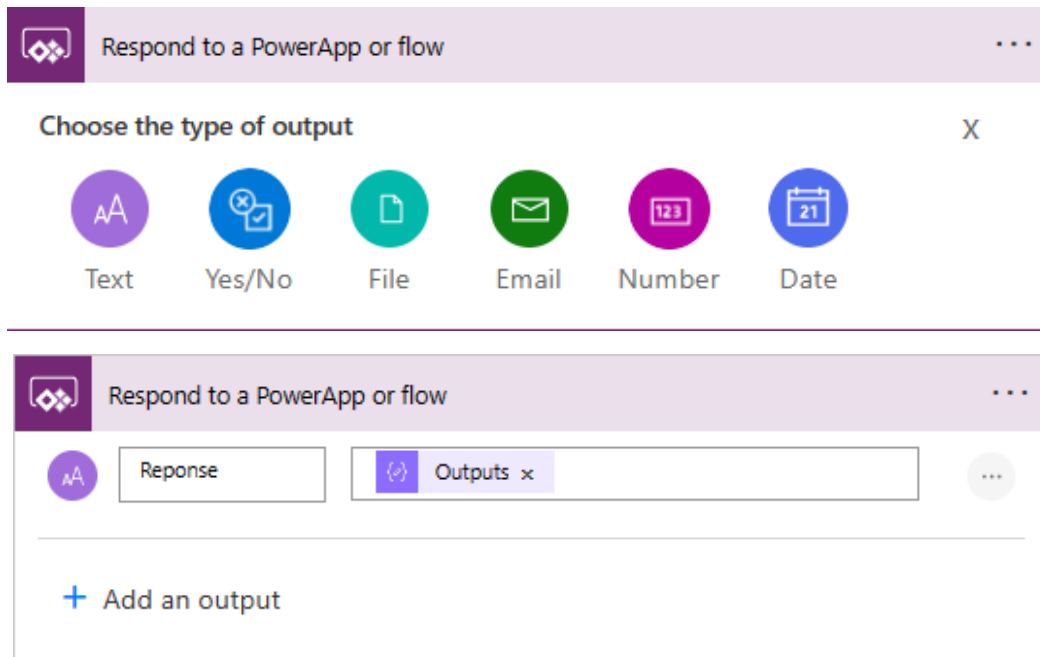

6. And store these parameters in the corresponding Compose actions:

7. Add new **Compose** action, name it **response,** and set the Inputs to **Team created**:
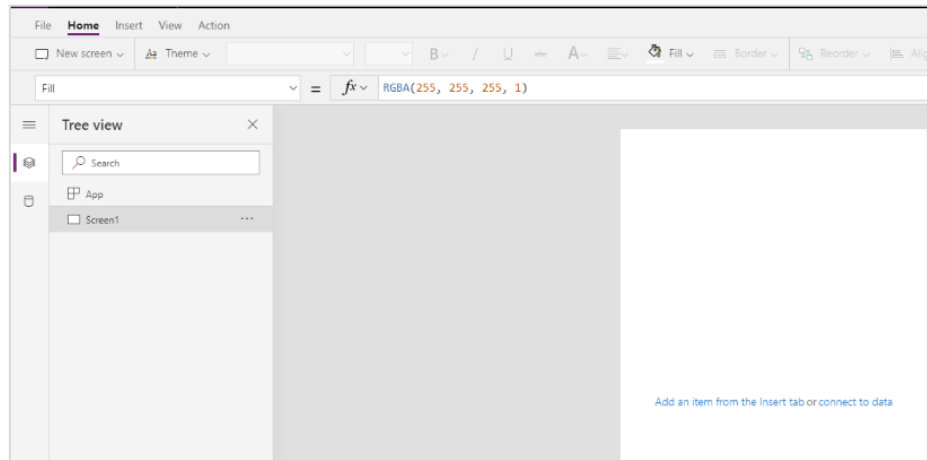

8. Add an action **Respond to a PowerApp or flow** that will return our response. Click Text as the output type:

9. Save your Flow.

10. Now you will create a **Power Apps application** that will invoke our flow. Go to **PowerApps.com**. Click on **Apps** menu and create an App
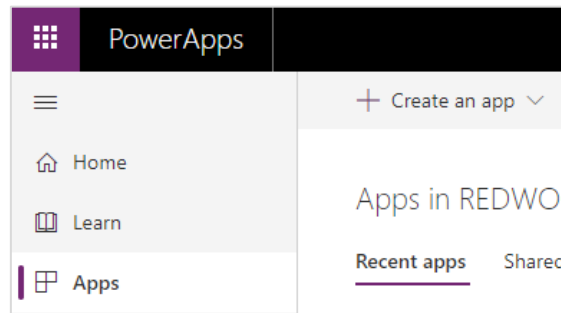
11. Select the **Canvas app**.
12. Click **Phone Layout**. The **PowerApps Designer** should show up:

13. Add 2 Text labels, two text Input, (rename them **TextInputTeamName** and **TextInputOwner**), and one button (rename it **btnCreateTeam**):
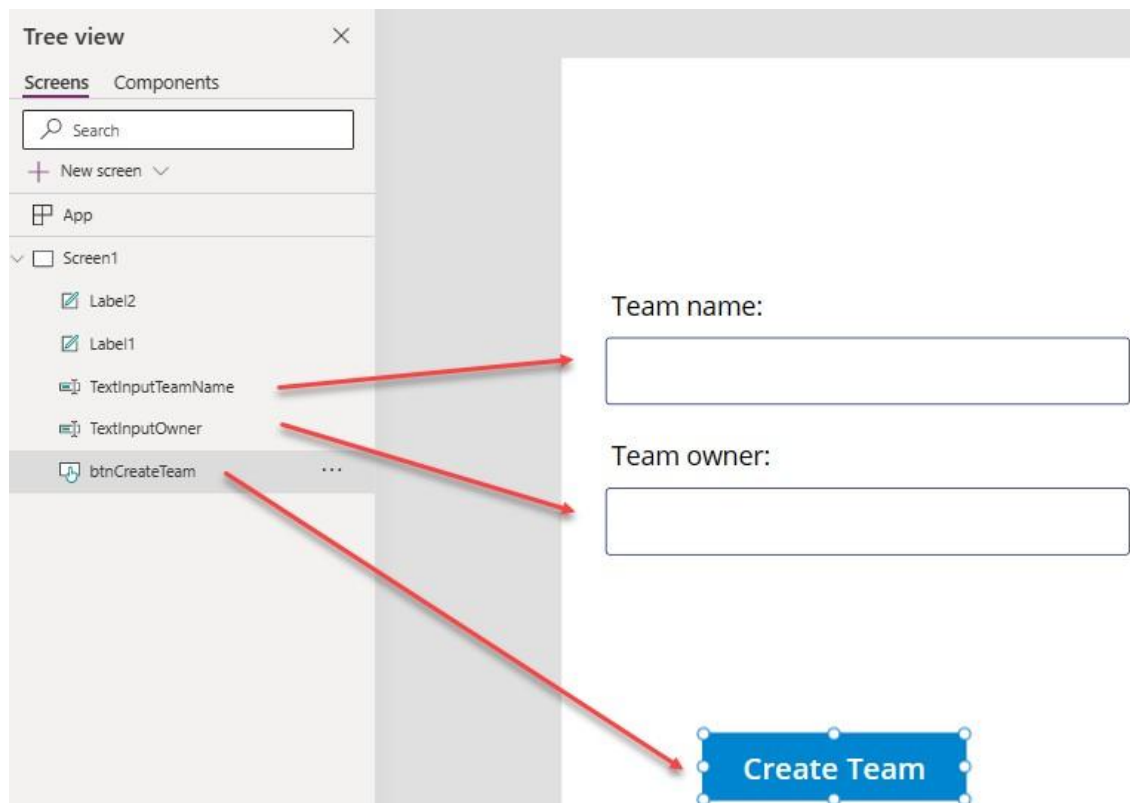
14. In the menu on the left, click the **Power Automate** icon:

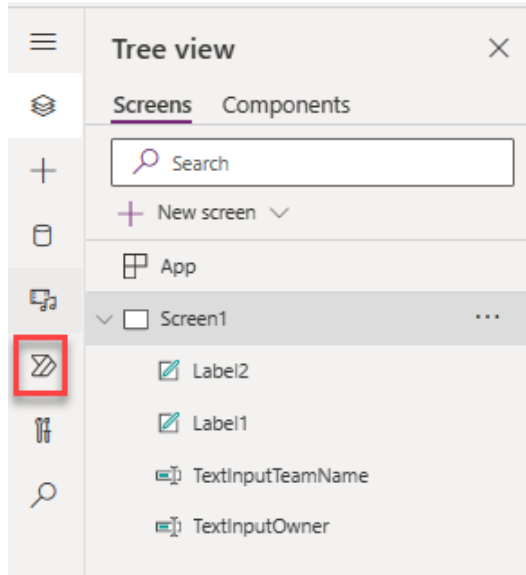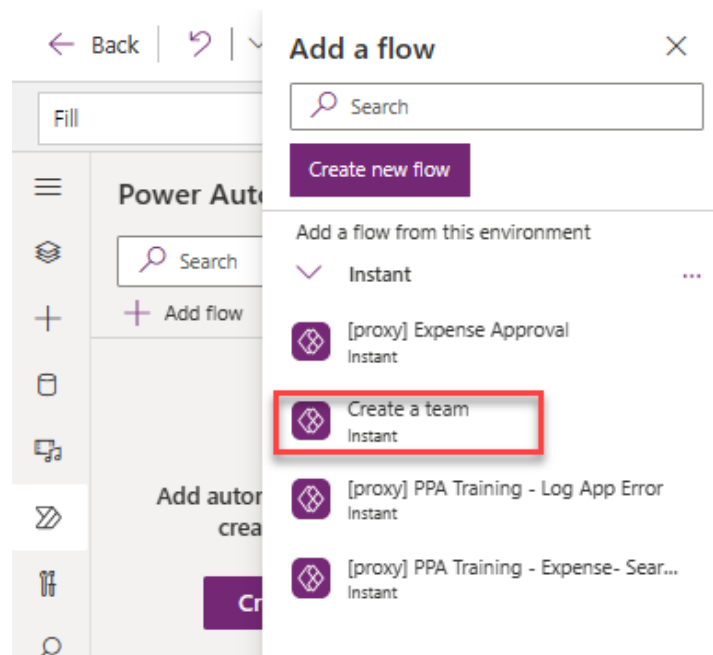15. Click **Add flow** to bring the list of flows starting with a Power Apps trigger:, and **select Create a team :**

16. In the OnSelect property of the button, call your flow:

Set (FlowStatus, Createateam.Run(TextInputTeamName.Text,TextInputOwner.Text))

If you are testing your code in a Dataverse for Teams environment, the code should be:

Set (FlowStatus, Createateam.Run(TextInputTeamName.Value,TextInputOwner.Value))

[Note:

If you are using a non US version of Power Apps, the comma in the formula should become a semicolon ; the code becomes **Set (Flowstatus; CreateTeamFromPowerApps.Run(TextInputTeam.Text; TextInputOwner.Text)) .**

]

17. The value the flow returns will be stored in a variable called **FlowStatus**.

| OnSelect | ∨ | = | *fx* ∨ | Set (flowstatus, CreateTeamFromPowerApps.Run(*TextInputTeam*.Text, *TextInputOwner*.Text)) |
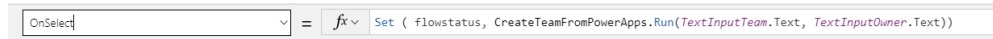
18. We will display the status in another label; add another label to the form:
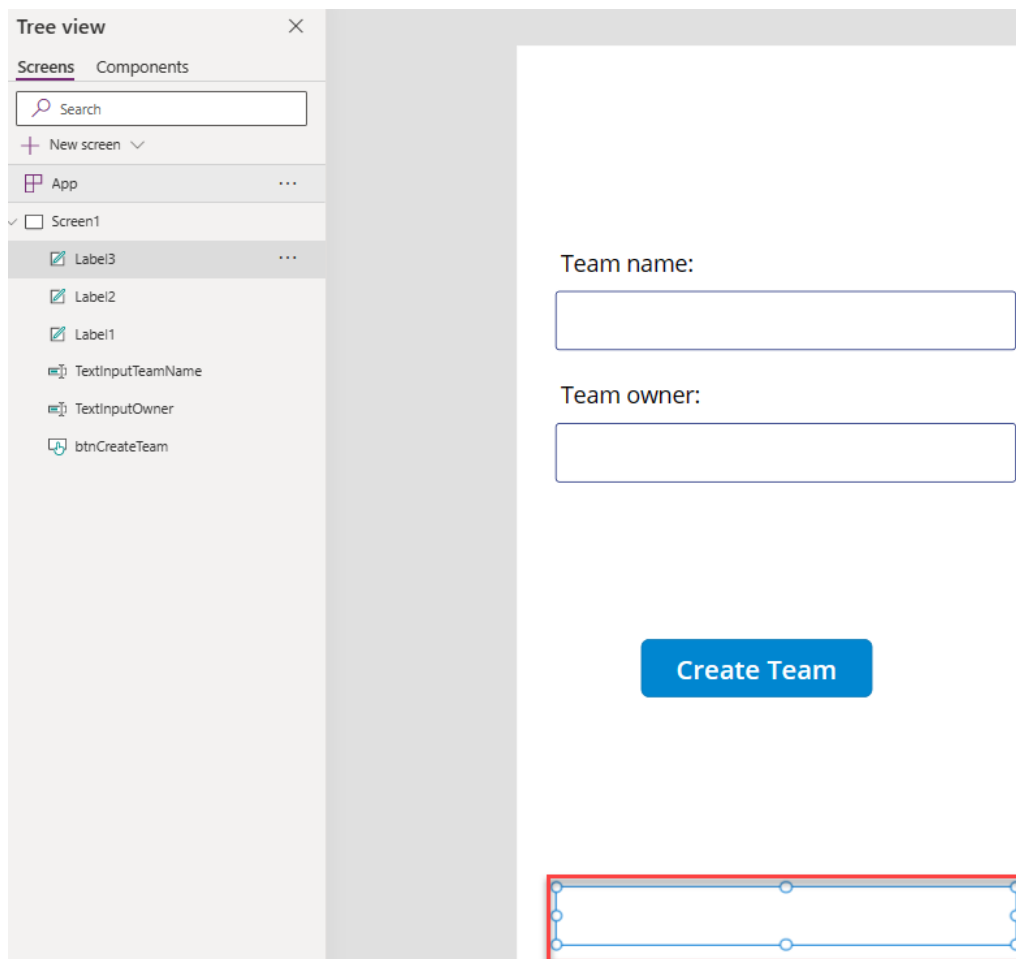
19. This new label Text property must be **FlowStatus.response**.
20. Run the Application, click the Submit button, and you should see the returned value:

Team name:

doctor

Team owner:

doctorflow@drflow.com

Create Team

Team created

We need your feedback

Do you want to report an issue or suggest something? We need your feedback:
https://github.com/Power-Automate-in-a-day/Training-by-the-community/issues