 Microsoft Power Automate Desktop

Lab 03: Variable types and data manipulation

Hands-on lab step-by-step

July 2024

Microsoft Power Automate Desktop – Advanced Workshop

Contents

[Microsoft Power Automate Desktop 1](#_Toc172808324)

[Goals for this lab 1](#_Toc172808325)

[Prerequisites 1](#_Toc172808326)

[Exercise: Get familiar with variables in Power Automate Desktop 2](#_Toc172808327)

[Terms of Use 13](#_Toc172808328)

# Microsoft Power Automate Desktop

This lab is subject to the Terms of Use found at the end of this document.

## Goals for this lab

|  |  |
| --- | --- |
| After this lab you will be able to:   * Understand the different types of variables in Power Automate Desktop and the available actions to manipulate them | The time to complete  this lab is [40] minutes. |

## Prerequisites

The labs have been designed so if you have access to a Microsoft Power Automate Desktop trial, you can get started from most labs without having to complete the previous module to be able to move forward. However, for the best experience that shows the features and functionality that is possible within the product, it is recommended you have completed specific modules before starting some of the labs.

For Lab 03: Variable types and data manipulation, you need:

* A computer with internet access.
* The application Power Automate Desktop installed in your computer. If you don’t have the application installed, please download it here: <https://go.microsoft.com/fwlink/?linkid=2102613>
* Be able to log into your corporate tenant.

## Exercise: Get familiar with variables in Power Automate Desktop

### Task 1: Log into Power Automate Desktop

1. Open the **Power Automate Desktop** app on your computer
2. Log into the application using your corporate account
3. If you don’t have a Power Automate Premium license, start a trial by clicking on the **Go Premium** button at the top right corner of the application

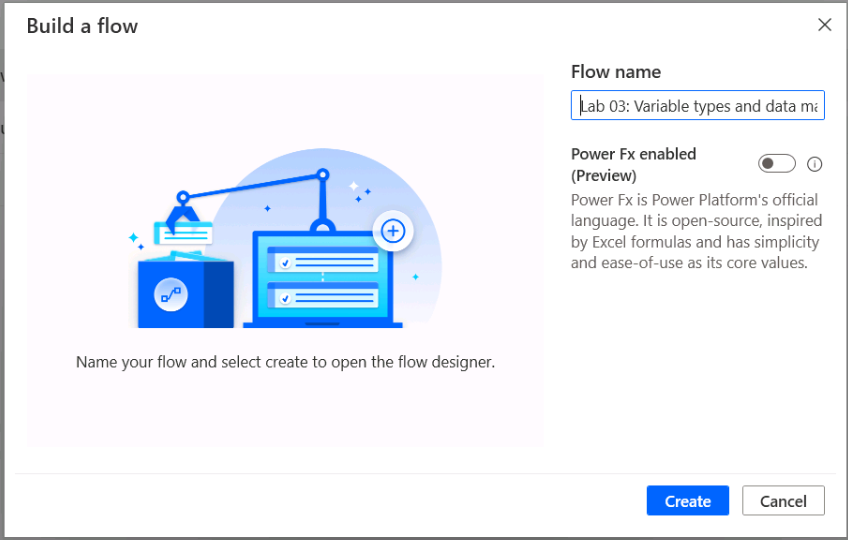


### Task 2: Create a desktop flow and login to a website using object variables

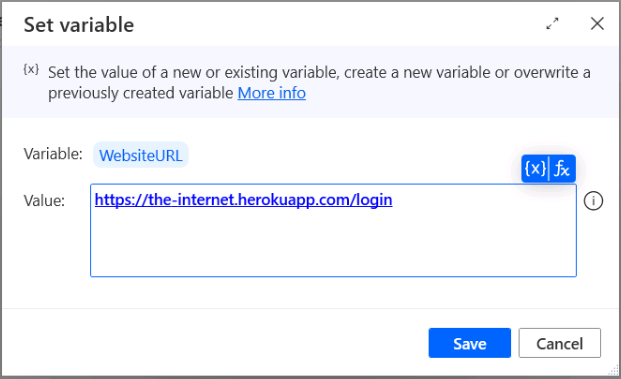
1. Make sure you create your automations in your own **Personal Development environment**. If you don’t have one, follow the steps here: [Get your developer environment - Power Apps | Microsoft Learn](https://learn.microsoft.com/en-us/power-apps/maker/maker-create-environment#create-your-own-developer-type-environment)

**Special note for Walmart 💡:** Power Platform administrators have blocked the option to create Personal Development Environments in the tenant. For this training, you can use the default environment titled **Walmart Store Inc.**, but in production scenarios, it is a best practice to develop the automations on a dedicated environment. Contact your team leader for more information about your dedicated environment.

1. Click on **+ New Flow** and create a flow named Lab 03: Object variables and webpage login Add your name at the end of the flow to recognize it easily.
2. Turn the **Power Fx enabled** toggle off. Click on **Create**

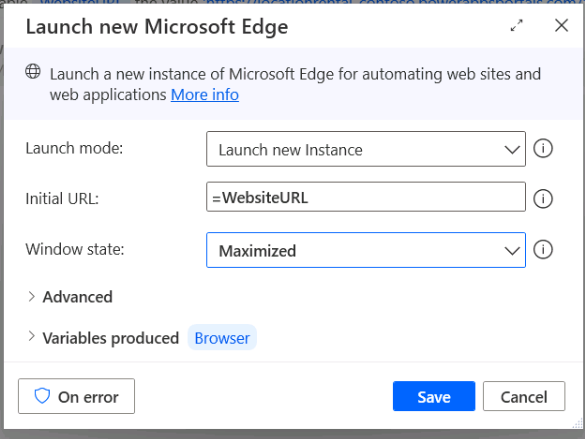


1. Once on the flow designer, look for action **Set variable**
2. Create a new variable named WebsiteURL and assign the value https://the-internet.herokuapp.com/login



1. Add the action Launch New Microsoft Edge (or Launch New Chrome) and configure it as shown next. Click **Save** after configuring the action:

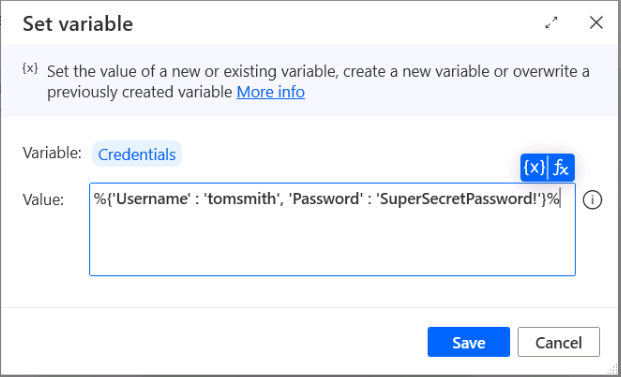
* **Launch mode:** Launch new instance
* **Initial URL:** %WebsiteURL%
* **Window state:** Maximized



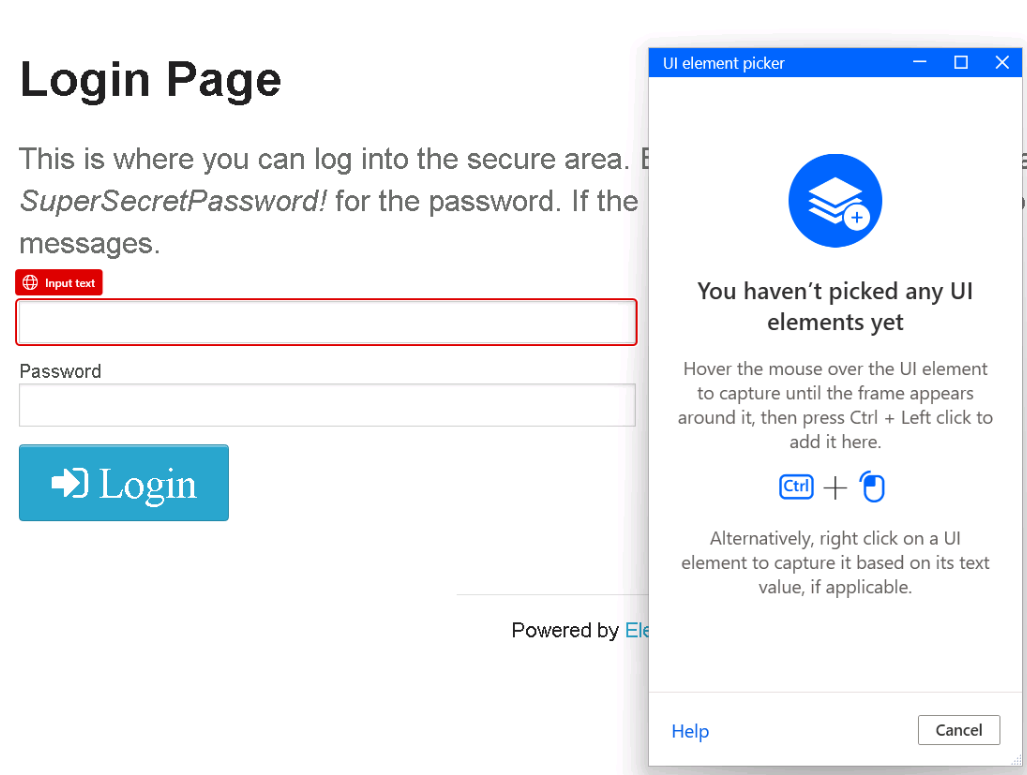
1. **Run** the flow and make sure it navigates to the desired page. Keep the page open and return to the flow designer.
2. Add a **Set variable** action and create a new variable named Credentials. On the value, create a custom JSON object with the following properties:

%{‘Username’ : ’tomsmith’, ‘Password’ : ‘SuperSecretPassword!’}%

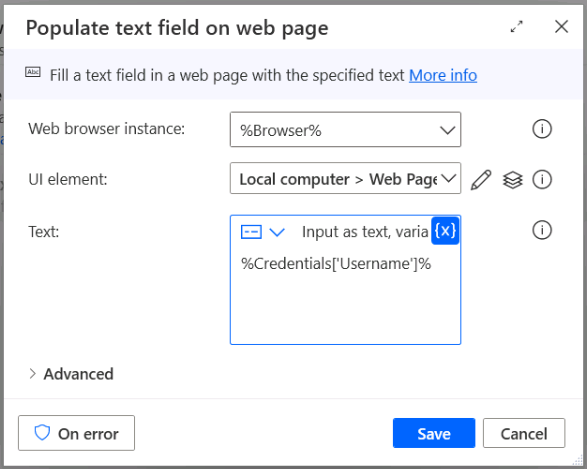
**Warning ⚠️:** If you copy and paste the above expression, make sure to replace all single quote characters



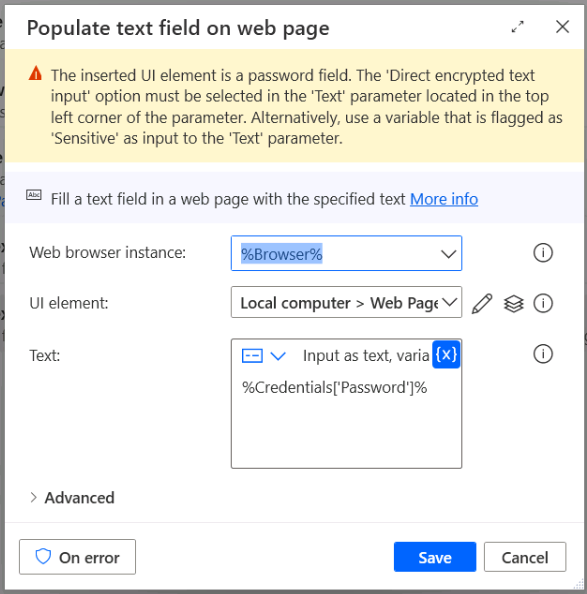
1. Add a **Populate text field on webpage** action and configure a new UI element. Select the field on the webpage to write the username



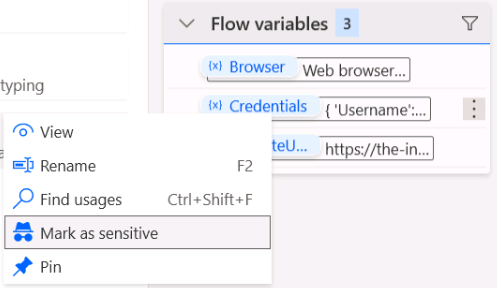
1. On the **Text** parameter of this action, type the formula %Credentials[‘Username´]%. Click **Save**



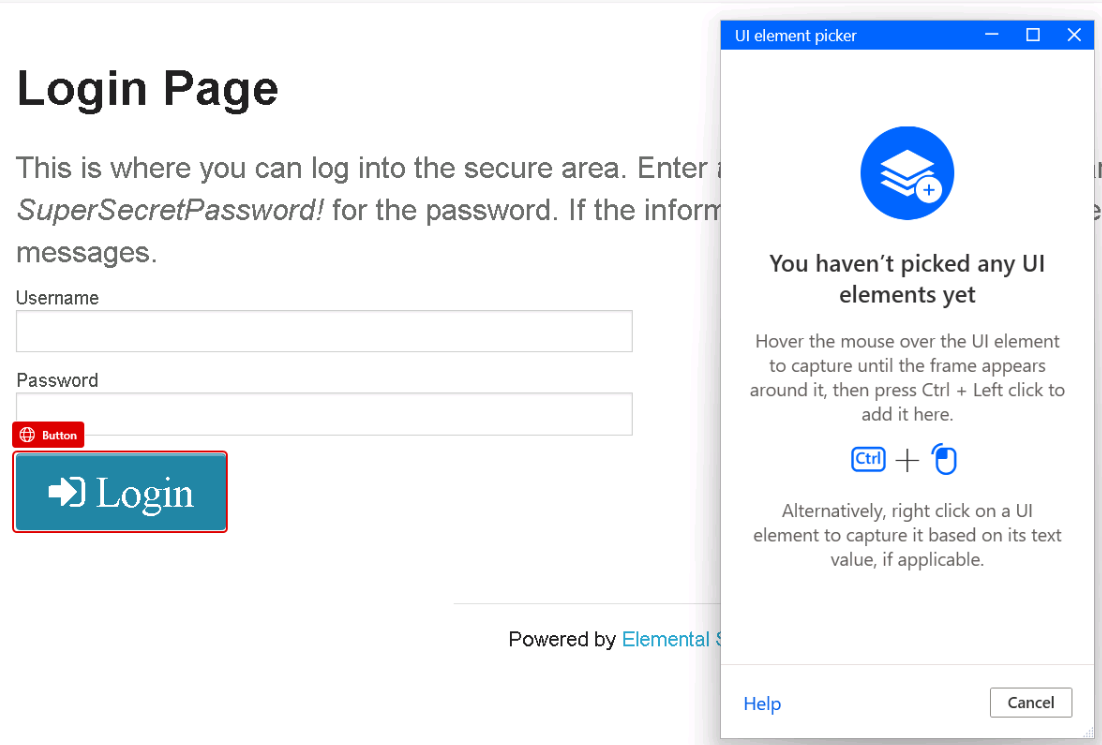
1. Repeat steps 9 and 10 to populate the Password field on the webpage.



1. You will see a warning message above your action. To make the password safe, go to the left side panel and right-click on the Credentials variable. Activate the **Mark as sensitive** option.

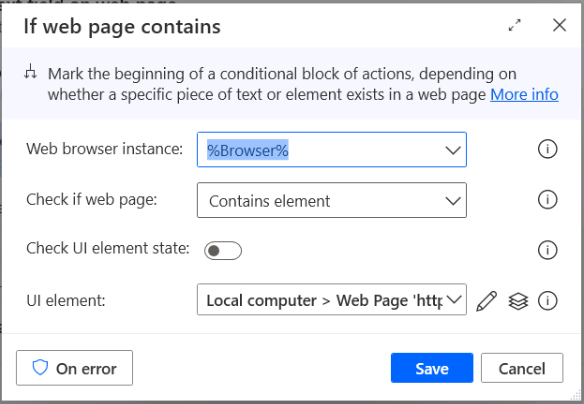


1. Add the action **Click link on webpage** to your flow. Add the UI element for the Login button of the webpage.

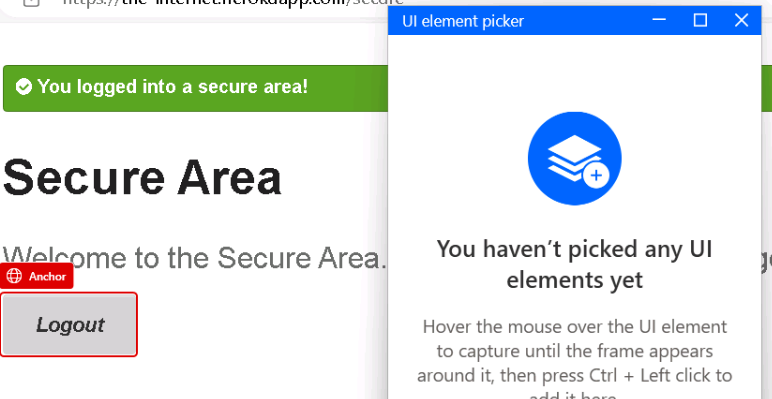


1. Close your browser instance and **Run** your flow. Ensure the process is logging into the website successfully.
2. Add the action **If webpage contains** to the flow. Configure it as shown:

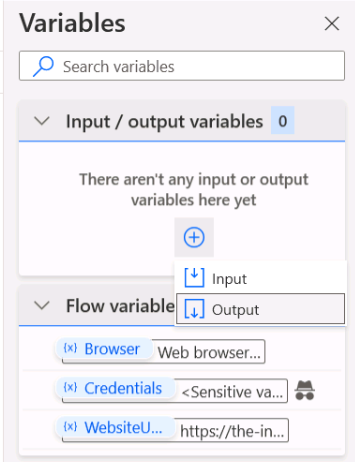
* **Web browser instance:** %Browser%
* **Check if webpage:** Contains UI element
* **UI element:** Add the UI element of the Logout button on the webpage



1. Inside the conditional block, add the action **Wait** and create a 5 second delay. After this action, add **Click link on webpage** and select the UI element to **Logout**



1. Navigate to the Input/Output variables section in your flow. Create a new output variable.



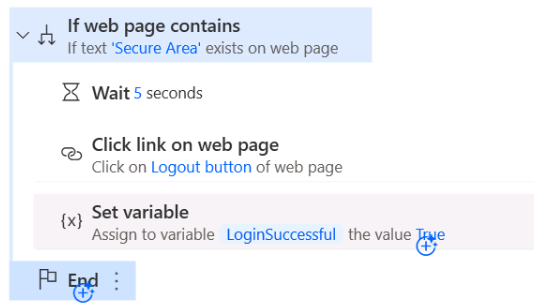
1. Configure the output variable parameters as shown:

* Variable name: LoginSuccessful
* Data type: Boolean
* External name: LoginSuccessful
* Description: (Blank)

A screenshot of a computer

Description automatically generated

1. Add an action to **Set a variable** in your conditional block. Set the variable %LoginSuccessful% to the value %True%



1. Add an **Else** block to the flow
2. Copy the **Set variable** action and change the %LoginSuccessful% value to %False%

A text box with a plus and a cross

Description automatically generated

1. **Run** your flow and inspect the value of the LoginSuccessful variable. To test an unsuccessful login case, try to change the value of the password on the Credentials variable and run the flow again.
2. It is a recommended practice to close the browser at the end of your automation. Add a **Close web browser** action to the end of your flow.

**Note 💡:** It is normal for the flow to take a few seconds to complete the Close web browser action after the browser window disappears.

### Task 3: Interact with Datatable variables

1. Download the following files from the GitHub repository. Keep track of the location of these files:

* [PowerAutomateDesktopTraining/Employees Group 01.xlsx at main · mcoloradodevs/PowerAutomateDesktopTraining · GitHub](https://github.com/mcoloradodevs/PowerAutomateDesktopTraining/blob/main/Employees%20Group%2001.xlsx)
* [PowerAutomateDesktopTraining/Employees Group 02.xlsx at main · mcoloradodevs/PowerAutomateDesktopTraining · GitHub](https://github.com/mcoloradodevs/PowerAutomateDesktopTraining/blob/main/Employees%20Group%2002.xlsx)

1. Click on **+ New Flow** and create a flow named Lab 03: Datatable variables Add your name at the end of the flow to recognize it easily.
2. Add a **Launch Excel** action and configure the following parameters:

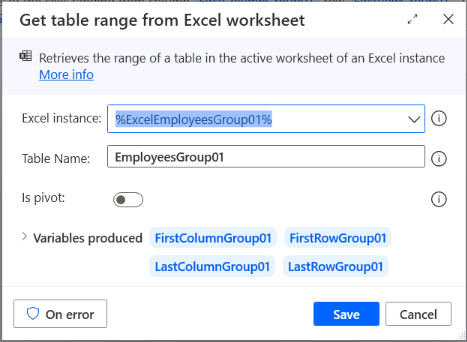
* Launch Excel: And open the following document
* Document path: The path of the document ‘Employees Group 01.xlsx’. For example: C:/Users/Admin/Desktop/Employees Group 01.xlsx
* Make instance visible: *Disabled*
* Variables produced: ExcelEmployeesGroup01

A screenshot of a computer

Description automatically generated

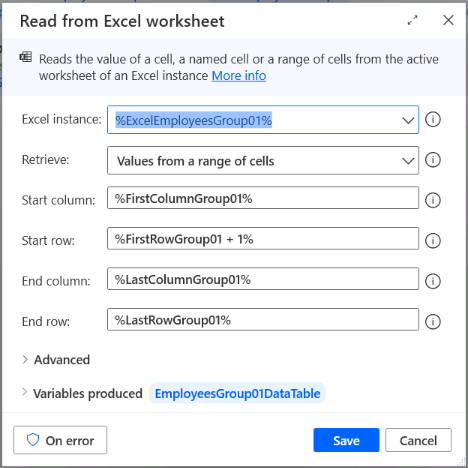
1. Add a **Get table range from Excel worksheet** action and configure it as shown:

* Excel instance: %ExcelEmployeesGroup01%
* Table name: EmployeesGroup01
* Variables produced: FirstColumnGroup01 FirstRowGroup01 LastColumnGroup01 LastRowGroup01

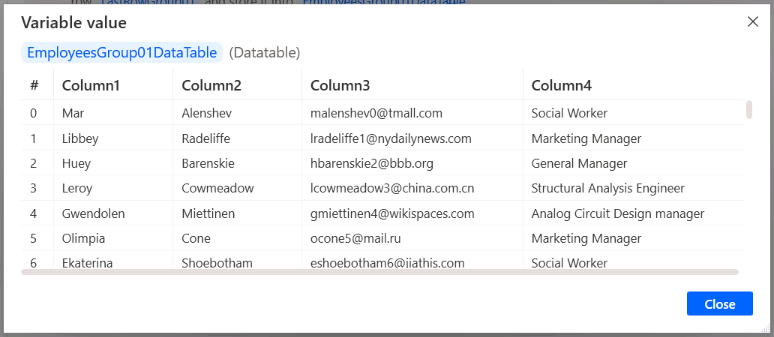


1. Add a **Read from Excel worksheet** action and configure it as shown:

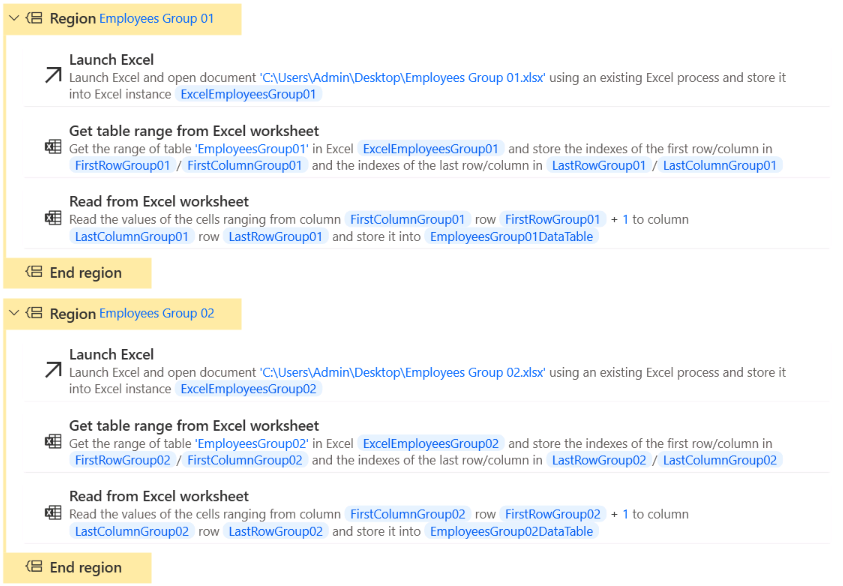
* Excel instance: %ExcelEmployeesGroup01%
* Retrieve: Values from a range of cells
* Start column: %FirstColumnGroup01%
* Start row: %FirstColumnGroup01 +1%
* End column: %LastColumnGroup01%
* End row: %LastRowGroup01%
* Variables produced: EmployeesGroup01DataTable



1. **Run** your flow and confirm the variable EmployeesGroup01DataTable now contains all values from your Excel worksheet

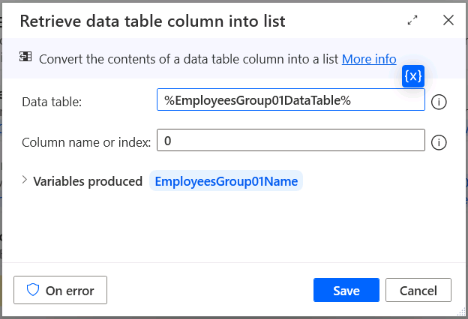


1. Repeat steps 3 to 6 to import the data from file **Employees Group 02**.

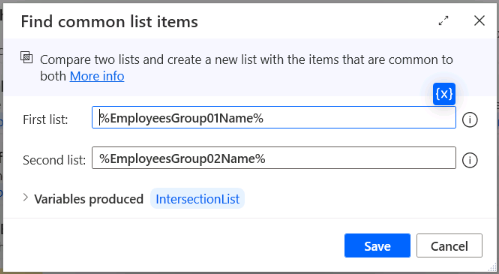


1. Add an action to **Retrieve data table column into list** and configure it as shown:

* Data table: %EmployeesGroup01DataTable%
* Column name or index: 0
* Variables produced: EmployeesGroup01Name



1. Repeat the previous step for the data table %EmployeesGroup02DataTable%
2. Add the action **Find common list items** and compare the lists EmployeesGroup01Name and EmployeesGroup02Name. Store the results into a variable named Common Employees



1. **Run** your flow and inspect the results in the Common Employees list. You should find a list of names that appear on both documents

## Terms of Use

By using this document, in whole or in part, you agree to the following terms:

### **Notice**

Information and views expressed in this document, including (without limitation) URL and other Internet Web site references, may change without notice. Examples depicted herein, if any, are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred. This document does not provide you with any legal rights to any intellectual property in any Microsoft product.

### **Use Limitations**

Copying or reproduction, in whole or in part, of this document to any other server or location for further reproduction or redistribution is expressly prohibited. Microsoft provides you with this document for purposes of obtaining your suggestions, comments, input, ideas, or know-how, in any form, ("Feedback") and to provide you with a learning experience. You may use this document only to evaluate its content and provide feedback to Microsoft. You may not use this document for any other purpose. You may not modify, copy, distribute, transmit, display, perform, reproduce, publish, license, create derivative works from, transfer, or sell this document or any portion thereof. You may copy and use this document for your internal, reference purposes only.

### **Feedback**

If you give Microsoft any Feedback about this document or the subject matter herein (including, without limitation, any technology, features, functionality, and/or concepts), you give to Microsoft, without charge, the right to use, share, and freely commercialize Feedback in any way and for any purpose. You also give third parties, without charge, the right to use, or interface with, any Microsoft products or services that include the Feedback. You represent and warrant that you own or otherwise control all rights to such Feedback and that no such Feedback is subject to any third-party rights.

### **DISCLAIMERS**

CERTAIN SOFTWARE, TECHNOLOGY, PRODUCTS, FEATURES, AND FUNCTIONALITY (COLLECTIVELY "CONCEPTS"),

INCLUDING POTENTIAL NEW CONCEPTS, REFERENCED IN THIS DOCUMENT ARE IN A SIMULATED ENVIRONMENT

WITHOUT COMPLEX SET-UP OR INSTALLATION AND ARE INTENDED FOR FEEDBACK AND TRAINING PURPOSES

ONLY. THE CONCEPTS REPRESENTED IN THIS DOCUMENT MAY NOT REPRESENT FULL FEATURE CONCEPTS AND MAY

NOT WORK THE WAY A FINAL VERSION MAY WORK. MICROSOFT ALSO MAY NOT RELEASE A FINAL VERSION OF SUCH

CONCEPTS. YOUR EXPERIENCE WITH USING SUCH CONCEPTS IN A PHYSICAL ENVIRONMENT MAY ALSO BE DIFFERENT.

THIS DOCUMENT, AND THE CONCEPTS AND TRAINING PROVIDED HEREIN, IS PROVIDED “AS IS”, WITHOUT WARRANTY

OF ANY KIND, WHETHER EXPRESS, IMPLIED, OR STATUTORY, INCLUDING (WITHOUT LIMITATION) THE WARRANTIES OF

MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND NONINFRINGEMENT. MICROSOFT DOES NOT

MAKE ANY ASSURANCES OR REPRESENTATIONS WITH REGARD TO THE ACCURACY OF THE RESULTS, THE OUTPUT THAT DERIVES FROM USE OF THIS DOCUMENT OR THE CONCEPTS, OR THE SUITABILITY OF THE CONCEPTS OR INFORMATION CONTAINED IN THIS DOCUMENT FOR ANY PURPOSE.