

# AUTOMATION PRO II – LABS

## Table of Contents

<b>Welcome to Automation Pro II.....</b>	<b>2</b>
Notes.....	2
<b>Lab 1 prep.....</b>	<b>3</b>
Create a Salesforce sandbox account.....	3
Connect to Salesforce.....	3
<b>Lab 1: Conditional actions.....</b>	<b>5</b>
Add industry data to Customer Contact List.....	5
Create a new recipe in Workato.....	6
Add an action in an app.....	7
Add steps for conditional actions.....	9
Configure actions for met conditions.....	10
Configure actions for unmet conditions.....	13
Save and test recipe.....	14
Review recipe output.....	14
Test recipe again.....	15
<b>Lab 2: Repeat actions.....</b>	<b>15</b>
Enter recipe edit mode.....	15
Add Repeat Action step.....	16
Choose source input list.....	16
Move steps into repeat action block.....	17
Address duplication error from Salesforce.....	18
Test recipe.....	19
Map Foreach datapills.....	20
Save and test recipe again.....	21
<b>Lab 3: Error monitoring &amp; handling.....</b>	<b>21</b>
Add Handle Errors step.....	21
Configure Monitor block.....	22

Configure On Error block.....	23
Induce error.....	24
Configure successful/unsuccessful messages.....	25
Save and test recipe to see the induced error.....	27
Verify output and test recipe again without Stop Job step.....	28
<b>Lab 4: Data modification and data mapping expanded.....</b>	<b>29</b>
Add data to Google Sheet.....	29
Map and transform new data.....	30
Save and test your recipe to view results.....	32
<b>Lab 5: Lists by Workato.....</b>	<b>32</b>
Add steps to accumulate lists.....	33
Add a step to email lists.....	37
Save and test your recipe to see results in your email.....	38
<b>Lab 6: Lookup tables.....</b>	<b>39</b>
Create a lookup table.....	39
Invoke your lookup table in a recipe.....	42
Map and transform customer data.....	43
Save and test your recipe.....	44
<b>Lab 7: Variables by Workato.....</b>	<b>44</b>
Create variables.....	44
Update variable: successful migrations count.....	47
Update variable: unsuccessful migrations count.....	48
Add migration tallies to email report.....	49
Save and test your recipe.....	50

# Welcome to Automation Pro II

Hello, and welcome to Automation Pro II -- the next step on your automation journey!

This course, designed for learners who have completed Automation Pro I, has helped thousands of learners advance their skills and knowledge in automation.

In this course, you will expand on lessons learned in Automation Pro I and create more recipes to practice your new skills. And, by the end of the course, you will be prepared to take the Automation Pro II certificates exam.

## Notes

We have designed labs to help you practice what you learn throughout this course. Each lab exercise will build on previous labs and lessons.

## Icons

-  The information icon will appear when there is important or additional information available about Workato.
-  The attention icon indicates important information or a required step.

## Lab 1 prep



A Google account and a Salesforce account are **required** for Automation Pro II labs.

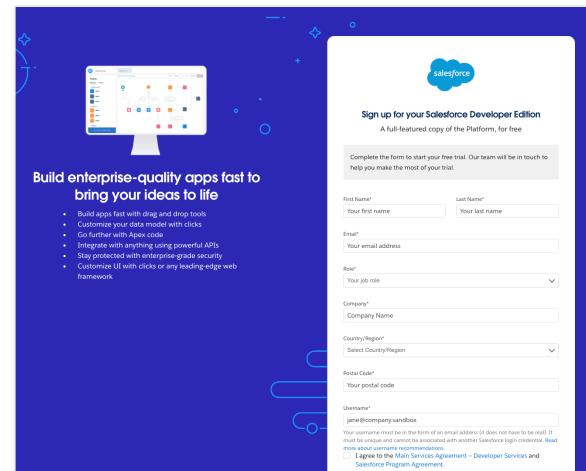
To complete the lab activities for this course, you will need access to a Google account and a Salesforce account. We recommend using sandbox instances of any applications you connect to during recipe design to avoid affecting production data.

In the Automation Pro I course, you were asked to create a Google sandbox account to access Google Sheets through Workato. We recommend using the same account for Automation Pro II labs. You will also need access to your original Customer Contact List Google Sheet from Automation Pro I.

### Create a Salesforce sandbox account

If you do not have a Salesforce sandbox account, **create** one at [developer.salesforce.com/signup](https://developer.salesforce.com/signup).

**Save** the username and password you create for your Salesforce account as you will need them to configure your Salesforce connection for this exercise.



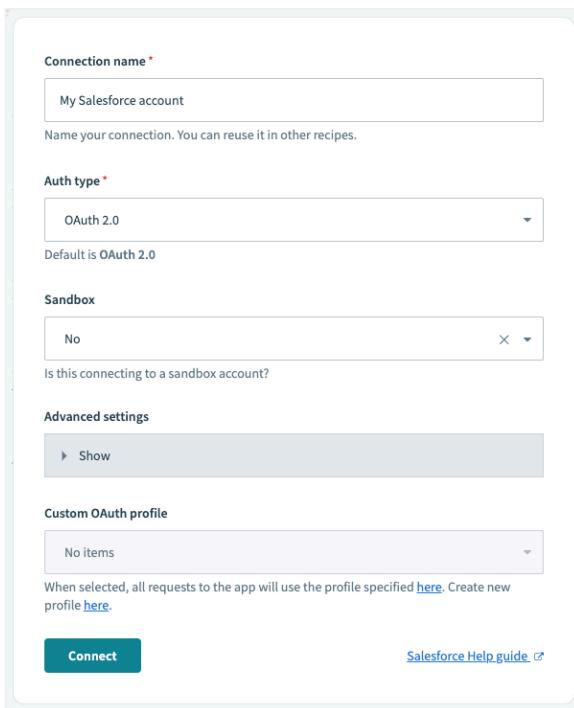
### Connect to Salesforce

Next, let's establish a connection to our new Salesforce sandbox.

To create a new connection, give the connection a name, such as **My Salesforce account**.

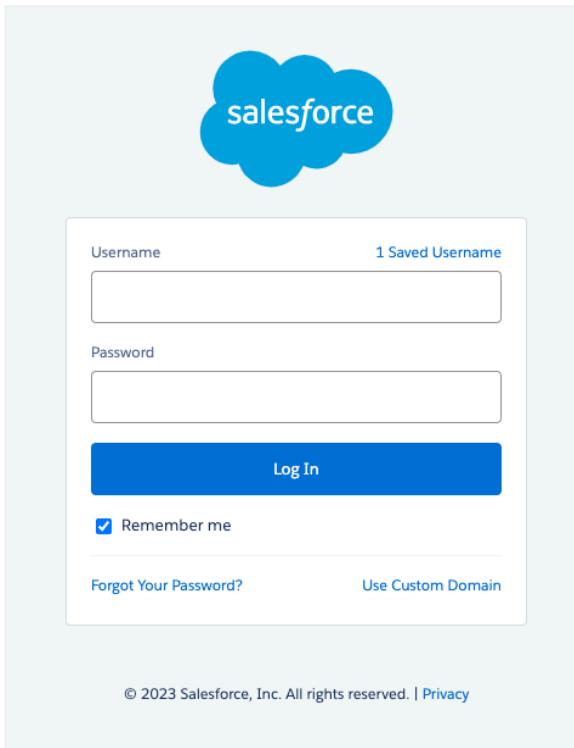
 Be sure to keep all other fields at their **default** settings.

Click **Connect**.



The screenshot shows the 'Create Connection' page for a 'Salesforce' connection. The 'Connection name' field contains 'My Salesforce account'. The 'Auth type' dropdown is set to 'OAuth 2.0'. Under 'Sandbox', the dropdown is set to 'No'. There is a 'Show' button for 'Advanced settings'. A 'Custom OAuth profile' section shows 'No items'. At the bottom are 'Connect' and 'Salesforce Help guide' buttons.

Then, input the username and password of the sandbox Salesforce account you just created to authenticate the connection.



The screenshot shows the Salesforce login page. It features the classic blue cloud logo. The 'Username' field has '1 Saved Username' next to it. Below it is a 'Password' field. A large blue 'Log In' button is centered. To its left is a 'Remember me' checkbox, which is checked. At the bottom are links for 'Forgot Your Password?' and 'Use Custom Domain'. The footer of the page includes the copyright notice '© 2023 Salesforce, Inc. All rights reserved.' and a link to 'Privacy'.

When prompted to allow access via Workato, click the **Allow** button.

- Perform segmentation on Customer Data Platform data
- Manage Customer Data Platform Identity Resolution
- Access Headless Forgot Password API
- Manage Customer Data Platform Calculated Insight data
- Perform requests at any time

Do you want to allow access for

(Not you?)

**Deny**

**Allow**

To revoke access at any time, go to your personal settings.

## Lab 1: Conditional actions

### Add industry data to Customer Contact List

First, we need to identify different industries for our customers.

In your Customer Contact List Google Sheet from Automation Pro I, add a column and name it **Industry**.

Populate industries for each customer, such as Finance, Retail, and Transportation. You can copy the industries in the following list:

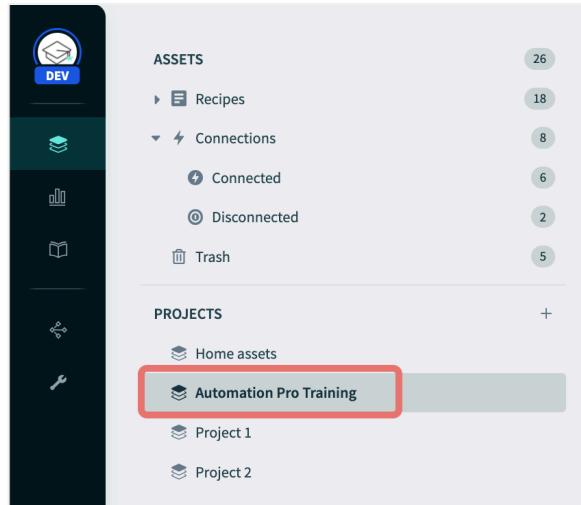
- Finance
- Transportation
- Retail
- Healthcare
- Education
- Power
- Logistics
- Manufacturing

First Name	Last Name	Email Address	Industry
Tariq	Miah	tariq@company1.com	
Walter	Jenkins	wjenkins@_company2.com	
Francine	Thibault	fran@company3.com	

First Name	Last Name	Email Address	Industry
Tariq	Miah	tariq@company1.com	Finance
Walter	Jenkins	wjenkins@_company2.com	Transportation
Francine	Thibault	fran@company3.com	Retail
Avinash	Jain	avinash.jain@company4.com	Healthcare
Alessia	Zullo	alessia@company5.com	Education
Irma	Klein	klein@company6.com	Power
Sumida	Akihiro	sumida@company7.com	Logistics
Tunç	Kurtoglu	tunc@company9.com	Manufacturing
Aisha	Tinibu	tinibu@company10.com	Finance
Jeom	Changmin	jeomchangmin@company11.com	Transportation
Reina	Perez	rp@company12.com	Retail
Aleksandr	Frolov	alex@company13.com	Healthcare
Bai	Tao	tao@company14.com	Education
Fearghas	Mahoneyi	ferg@company16.com	Power
Grayson	Rice	gray@company17.com	Logistics
Rita	Pedro	rpedro@company18.com	Manufacturing
Kelsey	Sutton	kelsey@company19.com	Finance
Sareema	Miah	sareema.miah@company20.com	Transportation
Sanjana	Dora	sanjaradora@company21.com	Retail

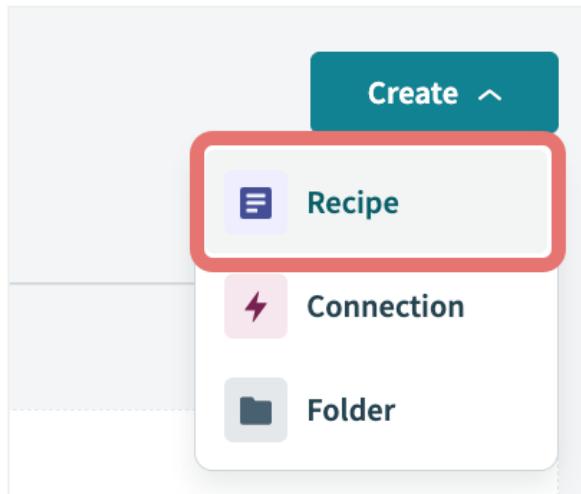
## Create a new recipe in Workato

To begin building your recipe, navigate into the training project you created earlier.



Click the **Create** button and click **Recipe**.

This will begin the recipe building process.



Give the recipe a relevant name such as **Migrate Google contacts to Salesforce by industry**.

**Set up your recipe**

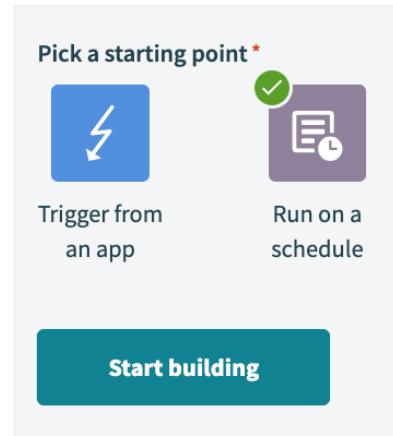
Name  
Migrate Google contacts to Salesforce by industry

Next, choose the **destination project** where you want to store your recipe.

Location\*  
Automation Pro Training

For this lab we are going to choose the project we're currently in: **Automation Pro Training**.

As we are using a scheduled trigger, let's select the starting point as **Run on a schedule** and click the **Start building** button.



You may be required to choose the time interval between triggers. Let's set our interval at **5 minutes** for this lab.

Time unit \*

Select an interval or custom schedule to specify cron expression.

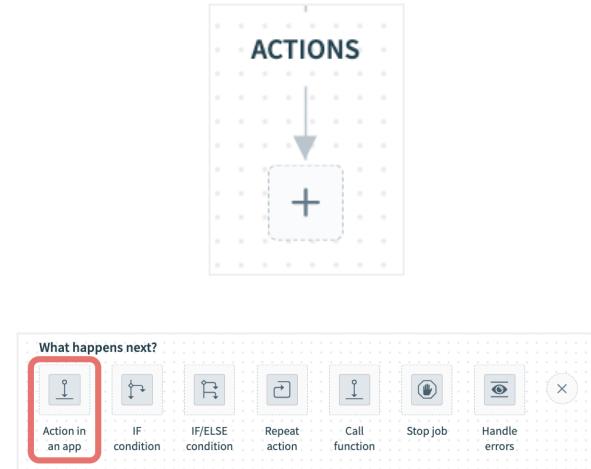
Trigger every \*

minutes

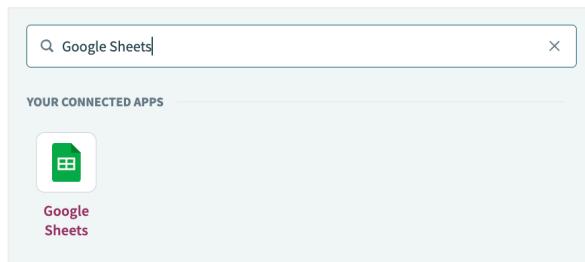
Define repeating schedule. Enter whole numbers only. This field can be set to a minimum of 5 minutes.

## Add an action in an app

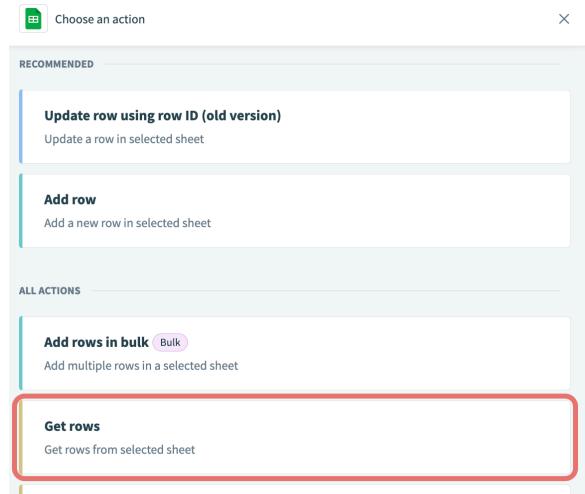
Next, click the **plus (+) icon** and choose **Action in an app**, as our action will involve the application, Google Sheets.



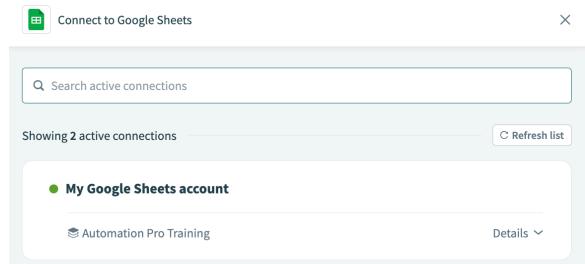
Search for and select the **Google Sheets** application.



For **Action**, choose **Get Rows**, because we want to review all rows for import eligibility.



Choose the Google Sheet connector you created in Automation Pro I.

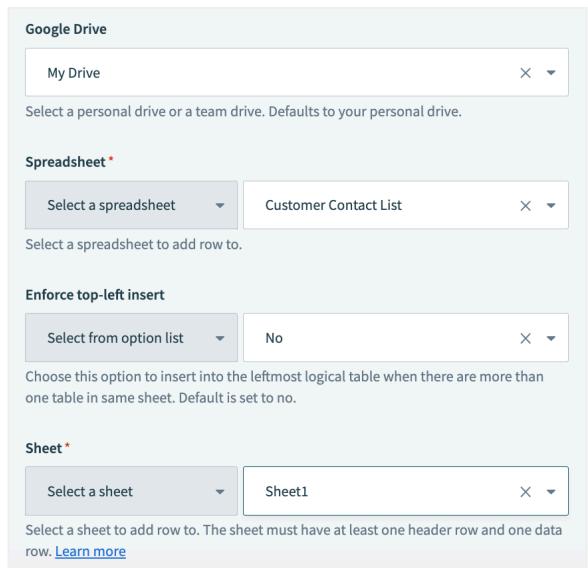


Once connected, we will finish setting up our Get Rows action.

For **Google Drive**, choose your drive from the dropdown list.

For **Spreadsheet**, choose the **Customer Contact List** spreadsheet we created during lab prep.

For **Sheet**, choose **Sheet1**.



The screenshot shows the configuration for a 'Get Rows' action. It includes fields for 'Google Drive' (set to 'My Drive'), 'Spreadsheet\*' (set to 'Customer Contact List'), 'Enforce top-left insert' (set to 'No'), and 'Sheet\*' (set to 'Sheet1'). Below these, a 'Range' field is set to '2:20'. A note at the bottom states: 'Provide range to retrieve rows from Sheet in the format Start row:End row. i.e. 2:2000 Note: Range should start from second row. Since first row is the header row.'

Set the range to **2:20** because we'll be starting our import at row 2.



The screenshot shows the 'Range' configuration field with 'ABC' selected in the first column and '2:20' in the second column. A note below says: 'Provide range to retrieve rows from Sheet in the format Start row:End row. i.e. 2:2000 Note: Range should start from second row. Since first row is the header row.'

## Add a conditional action

Click the **plus (+) icon** under Step 2 to start adding steps for conditional actions.

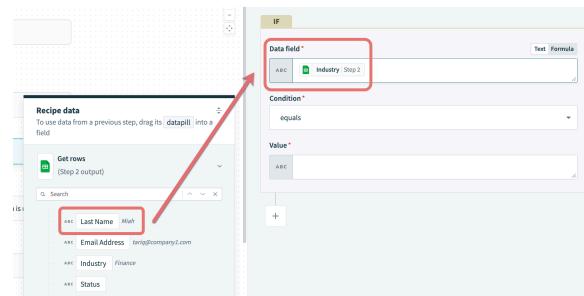



The screenshot shows the 'What happens next?' section with various options: Action in app, Recipe function, IF condition (which is highlighted with a red box), Repeat for each, Repeat while, Stop job, and Handle errors. Below this, a step labeled 'Setup your condition' is shown with a count of '3'.

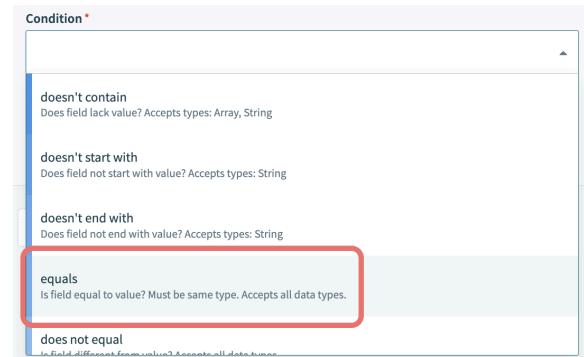
Choose **Step 3** to configure your **IF condition**.

For the **Data** field, let's choose **Industry** from our recipe data.

You'll find this object under the **Get Rows Step 2** output.



For **Condition**, choose **Equals** as we are looking for specific industries to import into Salesforce.



For **Value**, set your first industry to import. For the purposes of this lab, let's enter **Finance** as the industry.



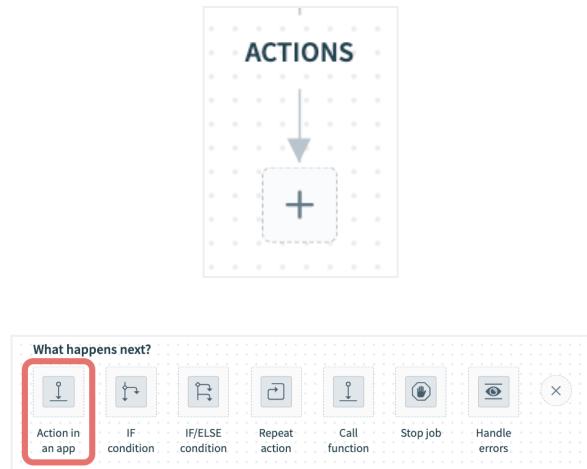
## Configure actions for met conditions

Choose **Step 4** to select an app and action to do when our condition is met.

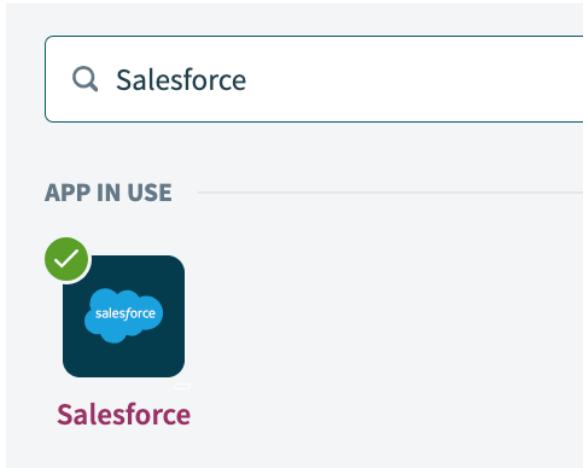


You'll notice the process looks very similar to how we've added actions in an app previously.

Click the **plus (+) icon** and choose **Action in an app**.



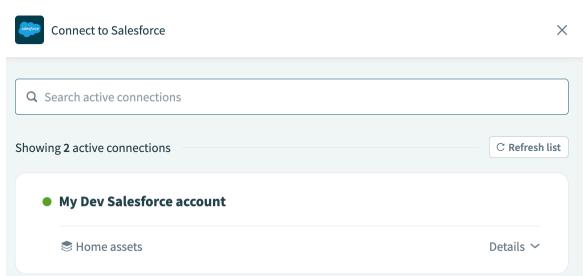
Search for and choose **Salesforce** as the app.



Choose **Create Record** for the action.



When prompted to choose a Salesforce connection, choose the Salesforce connector we created during lab prep.



For **Object**, search for and choose **Contact**.

Object\*

Contact

Account **Contact** Role

Case **Contact** Role

Contact

Contact Point Address

Contact last name is **required** by Salesforce so go ahead and map your **Last Name datapill** to the **Last Name field**.

Last Name \*

ABC

Last Name | Step 2

Map the **First Name datapill** to the **First Name field** and the as well as the **Email Address datapill** to the **Email Address field**.

First Name

ABC

First Name | Step 2

Email

ABC

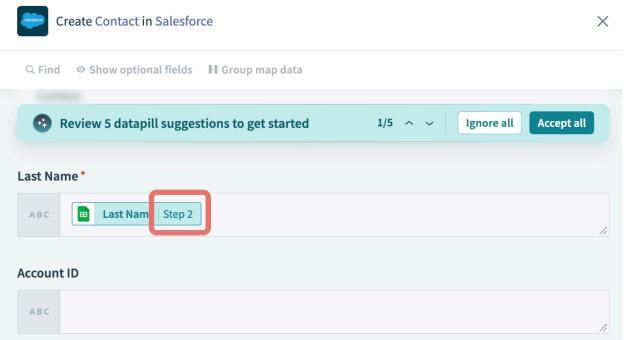
Email Address | Step 2

If you are unable to find the First Name or Email Address fields, **expand optional fields**.

+ 31 optional fields available

 Workato will suggest datapills that might match the fields we need to populate, such as Email Address or First Name.

If you decide to choose Workato suggestions, just ensure they are coming from the appropriate data source and recipe step.

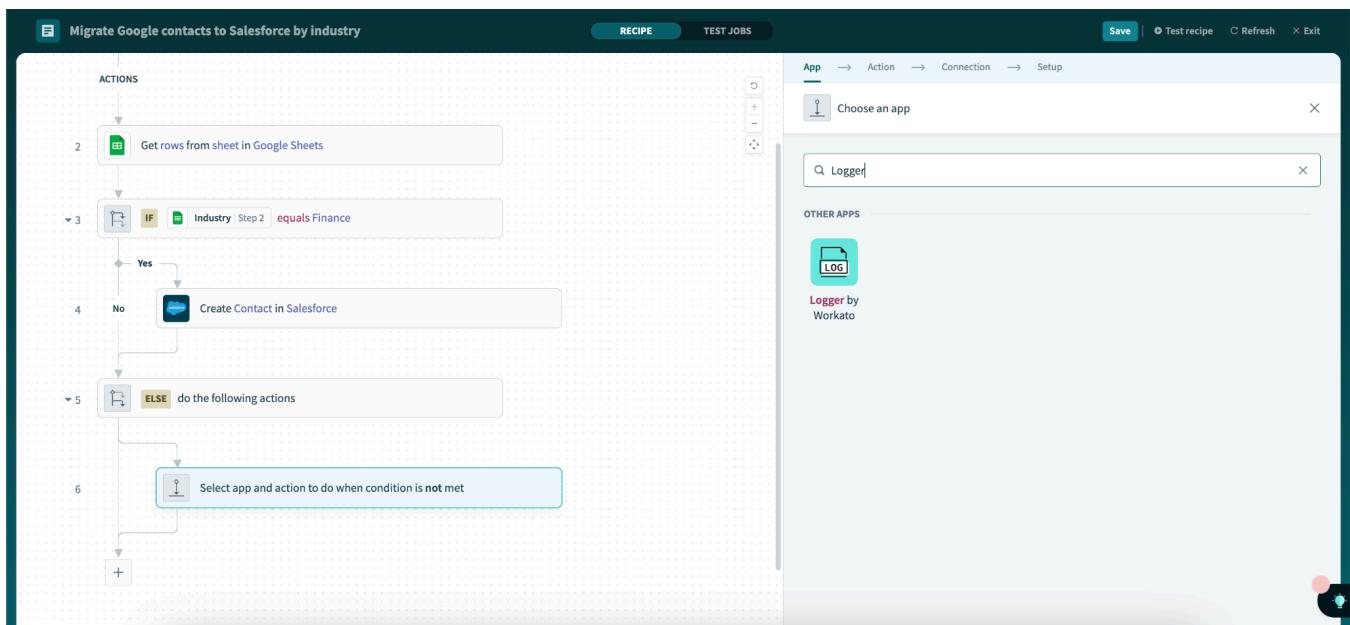


## Configure actions for unmet conditions

Click **ELSE condition** under **What happens next?**

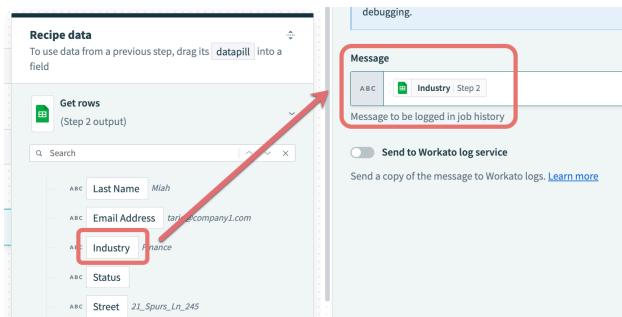
Choose **Step 6** to select an app and action to do when our condition is NOT met.

Search for and choose the **Logger by Workato** utility.



For **Message**, let's map **Industry**.

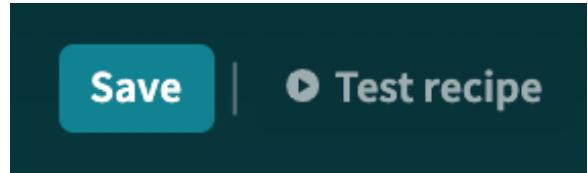
This will log any skipped industries to your job report.



## Save and test recipe

**Save** your recipe and run a **test** job to ensure your recipe is working properly.

You can then see your results in Salesforce in your **Contacts tab**.



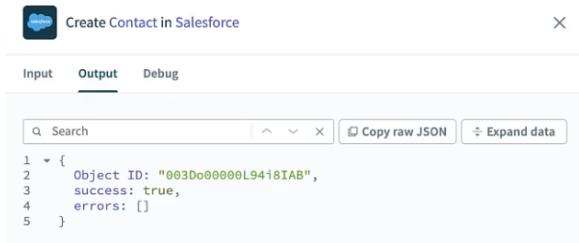
# Review recipe output

Back in Workato, let's take a quick look at the job report to see recipe output per step.

Click on **Step 2** and expand the data to see that Workato reviewed each row in your original Google Sheet.

Then, Workato determined that the first entry met the recipe condition.

Finally, Workato successfully created a contact in Salesforce based on information stored in your Google Sheet.

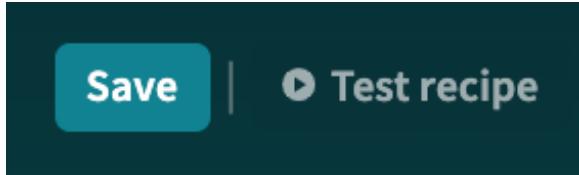


```
1  {
2     Object ID: "003Do00000L94i8IAB",
3     success: true,
4     errors: []
5 }
```

## Test recipe again

Now, let's **test** our recipe again.

You will see an error message. This is because this contact already exists in Salesforce.



The way the recipe is currently set up, the recipe will only process the **first item in the list**, unless the Repeat action is used to process all items in a list.

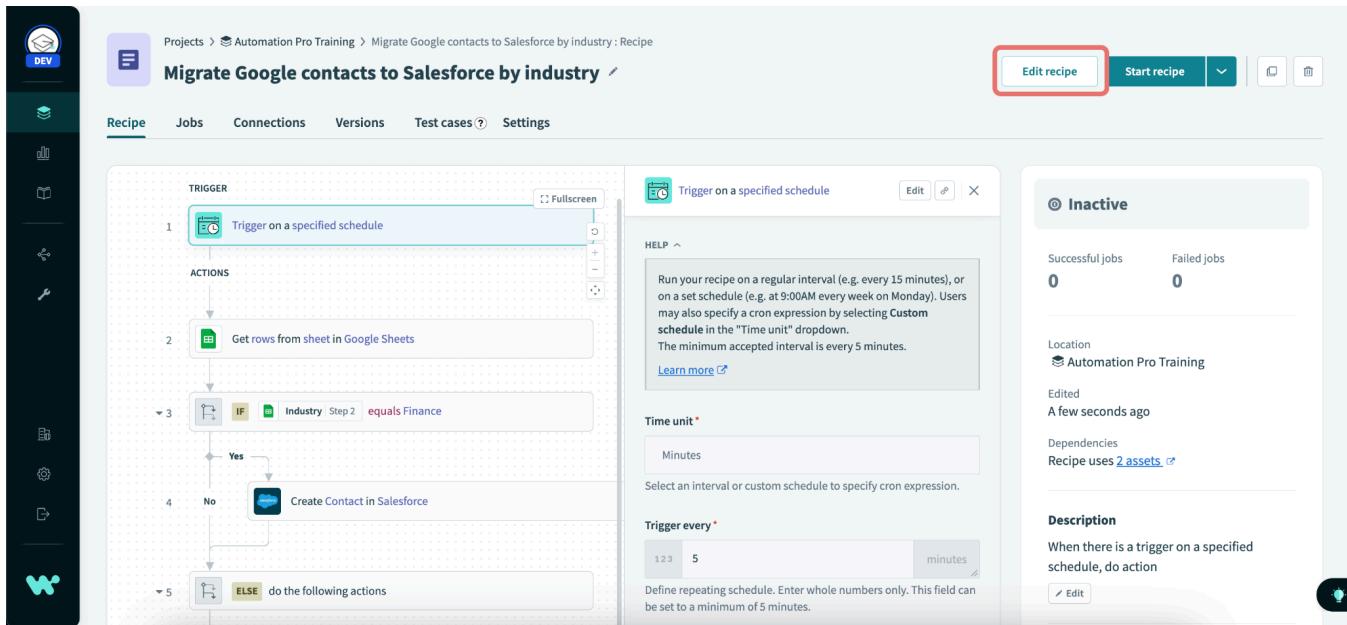
We'll go over the Repeat action next!

## Lab 2: Repeat actions

Let's return to the recipe from our previous lab and resolve our error message. We're going to redesign it to **process multiple rows at a time** by adding a repeat action.

### Enter recipe edit mode

If you are not already in recipe edit mode, click the **Edit** button when viewing recipe information.

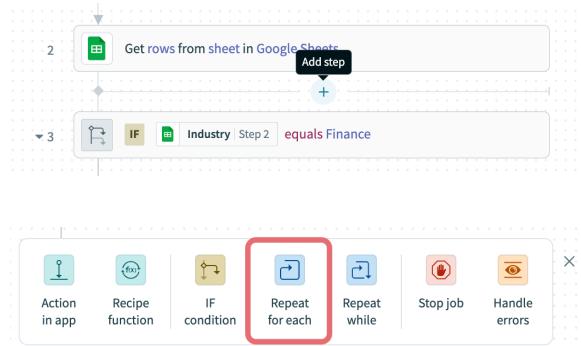


## Add Repeat Action step

Add a step between Steps 2 and 3.

-  You can add steps where needed by clicking the **plus (+) icon** that appears when you hover between steps.

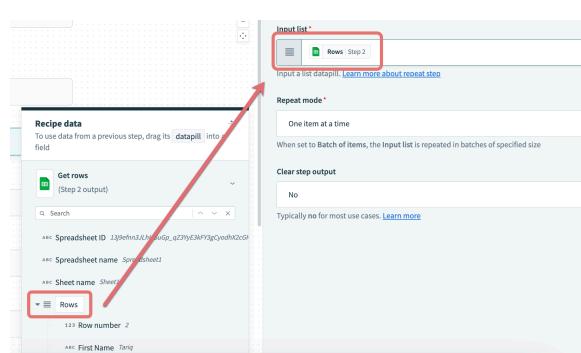
For **What happens next?** choose **Repeat for each**.



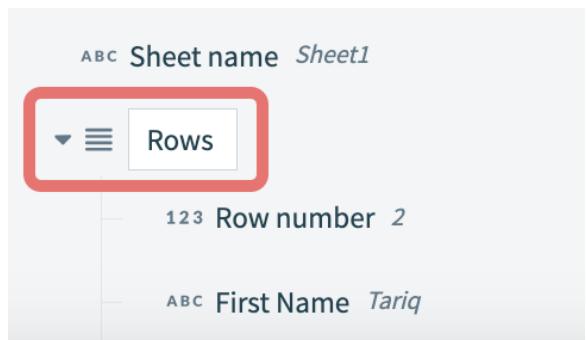
## Choose source input list

Locate the **Rows** list datapill under **Get Rows**, Step 2 output.

This directs Workato to **repeat each substep for each row** in your Google sheet.



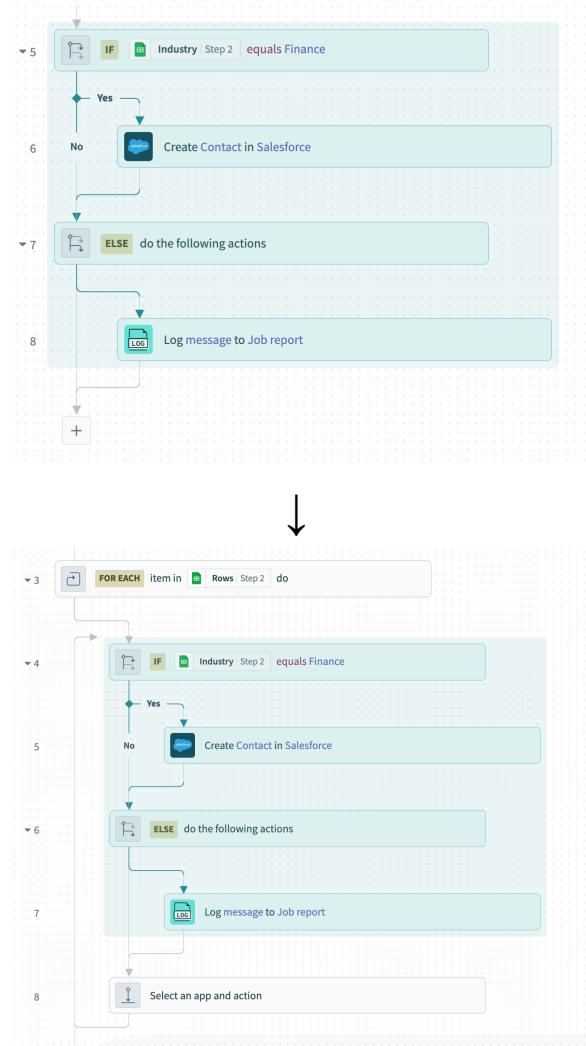
-  In recipe data, you can identify list datapills by the three horizontal bars that resemble a hamburger. This is called a **stack icon**.



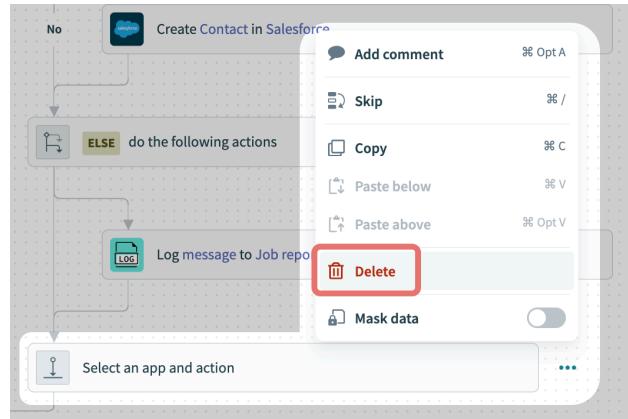
## Move steps into repeat action block

At this point, you could add repeatable substeps manually, but because we're redesigning our recipe for efficiency, we've made our work easier.

We're going to highlight our **IF/ELSE** conditional steps and drag them to **Step 4**.



Once these steps are in the correct location, we can **delete** any unnecessary steps.

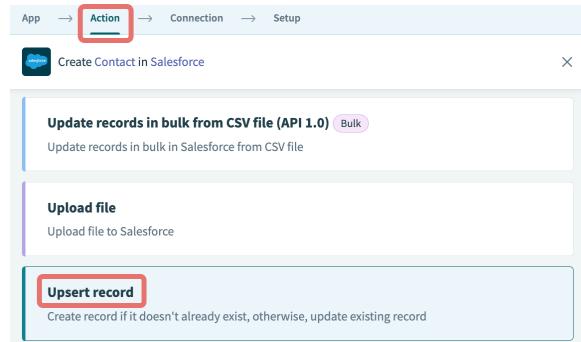


## Address duplication error from Salesforce

Now, let's fix our duplication error from our previous lab.

In **Step 5**, change the **Action** from **Create record** to **Upsert contact**.

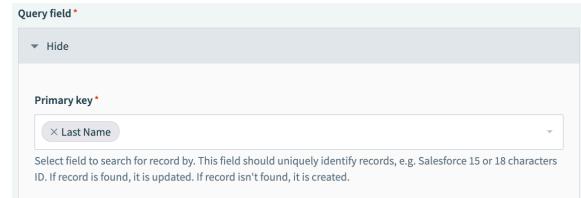
 Upsert is a combination of the words Insert and Update for Workato and is the process of inserting and/or updating data from one app into another app without creating duplicates.



### Why this is important

When we're syncing data between apps, we often run into situations where the data we're inserting from app 1 may already exist in app 2. We don't want duplicate data, so we need to check to see if it already exists. If it does, we update it; if it does not, we create it.

Choose **Last Name** as our **primary key** because the customer's last name will not be changing during our lab exercises.



Remap the datapills we'll use to create or update our Salesforce contact records: **Last Name**, **Email Address**, and **First Name** from **Step 2**.

**Last Name\***

ABC  Last Name | Step 2

Search for where Last Name has this value. If a record is created

**Account ID**

ABC

**Email**

ABC  Email Address | Step 2

**First Name**

ABC  First Name | Step 2

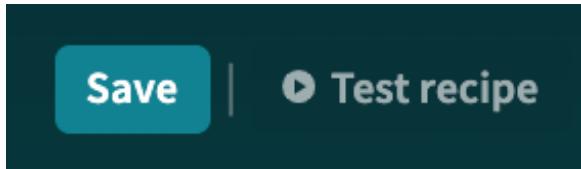
## Test recipe

Now, let's do a quick **test** of the recipe as is.

We've fixed our duplication error message, but we are still not seeing all of our finance contacts added to Salesforce.

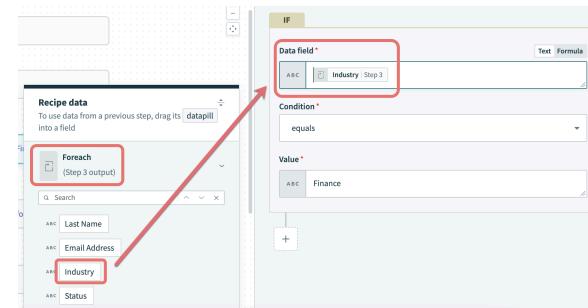
This is because the input to the repeat step is a list, and actions within a repeat block should use data output from the **repeat step's data tree**. This ensures that every item in the list is processed.

We'll fix our error by mapping datapills from the **Foreach** output data tree to ensure that values from each list item are used when the action is repeated.

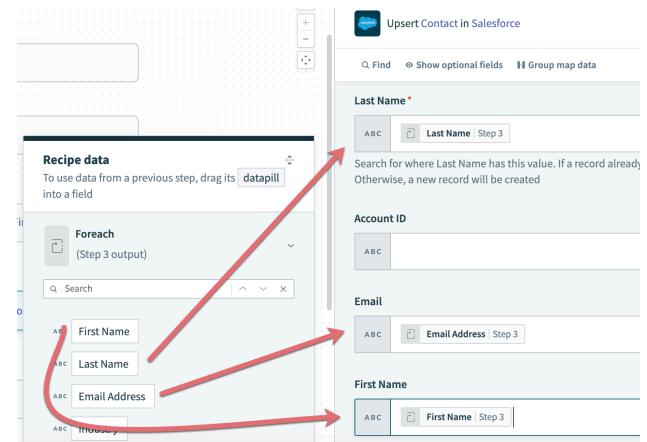


## Map Foreach datapills

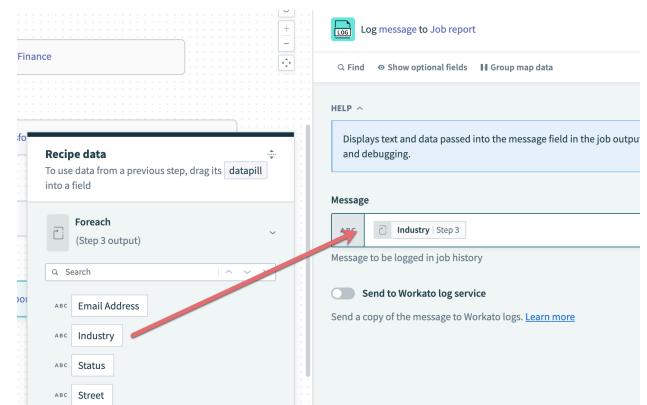
In Step 4, our **IF** step, replace the Industry datapill from Step 2 with the Industry datapill from Step 3, our **Foreach** step.



In our **Upsert Contact** step, Step 5, let's also replace **Last Name**, **Email Address** and **First Name** Step 2 datapills with corresponding Step 3 datapills.



Finally, in our **Log Message** step, Step 7, we'll replace the Industry datapill from **Step 2** with the Industry datapill from **Step 3**.

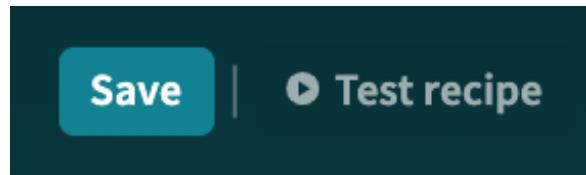


## Save and test recipe again

**Save** your recipe and **test** it again.

Once you see a successful job completion, return to Salesforce to see if your Contacts tab has been updated. You may have to refresh your Contacts tab.

You'll notice that Salesforce now contains all of the finance industry contacts from your original Google Sheet.



## Lab 3: Error monitoring & handling

Let's see how we can add error monitoring and handling logic to our recipe so we can log any recipe errors while a recipe is running.

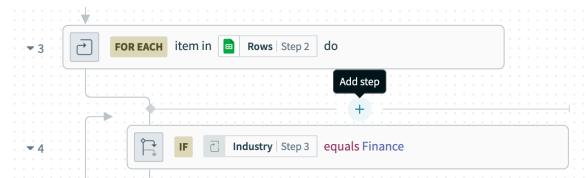
Return to the recipe from our previous lab and ensure you are in recipe edit mode.

### Add Handle Errors step

Hover between steps three and four to reveal a **plus (+) icon**.

Click the **plus (+) icon** to add a step after your Repeat step and before your first Conditional Action step.

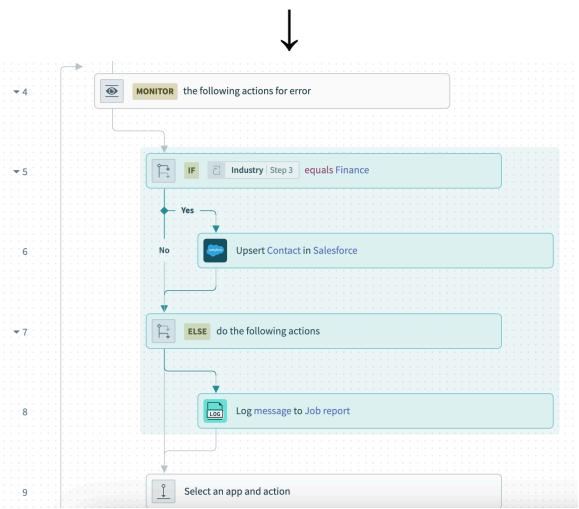
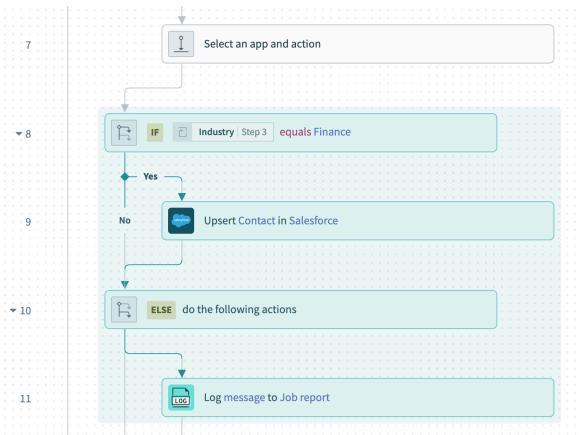
Choose **Handle Errors** to add a Monitor block and an On Error block to your recipe.



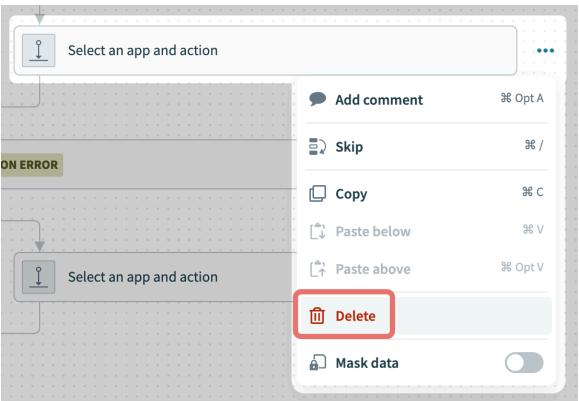
## Configure Monitor block

Highlight, then drag the **Conditional Action** steps (Steps 8 thru 11) into the Monitor block between **Steps 4 and 5**.

Now our recipe will monitor each iteration of the Repeat step.



Go ahead and **delete** the unnecessary Step 9 in the On Error block of steps.

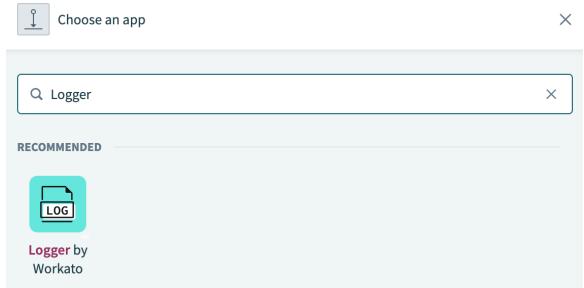


## Configure On Error block

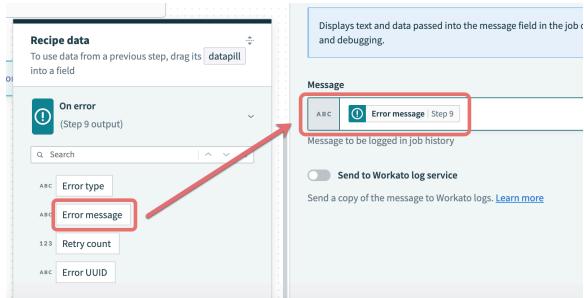
Locate **Step 10** to add an action in an app.

This will be the recipe's action should an error occur.

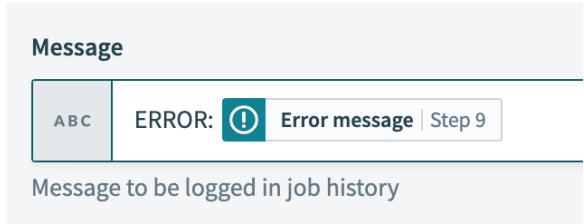
Search for and choose **Logger by Workato** as our app.



We want to log the type of error, so map the **Error Message** datapill from the On Error Step 10 output.

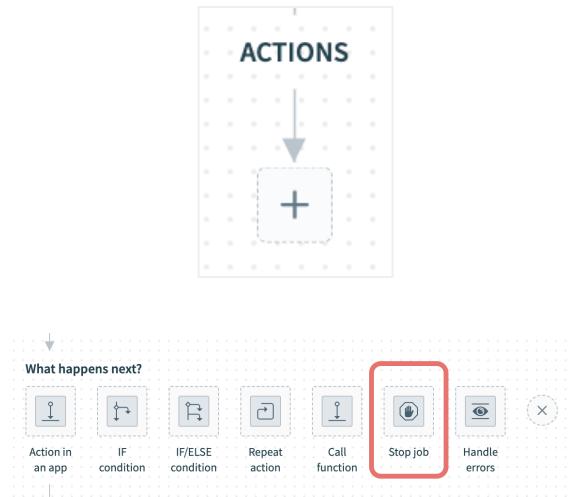


We could also customize our message by adding the word **ERROR** with a colon in front of the datapill.



**Add Step 11** within the On Error block to add an action to stop the recipe after an error has been logged.

To do this, click the **plus (+) icon** and choose **Stop Job**.



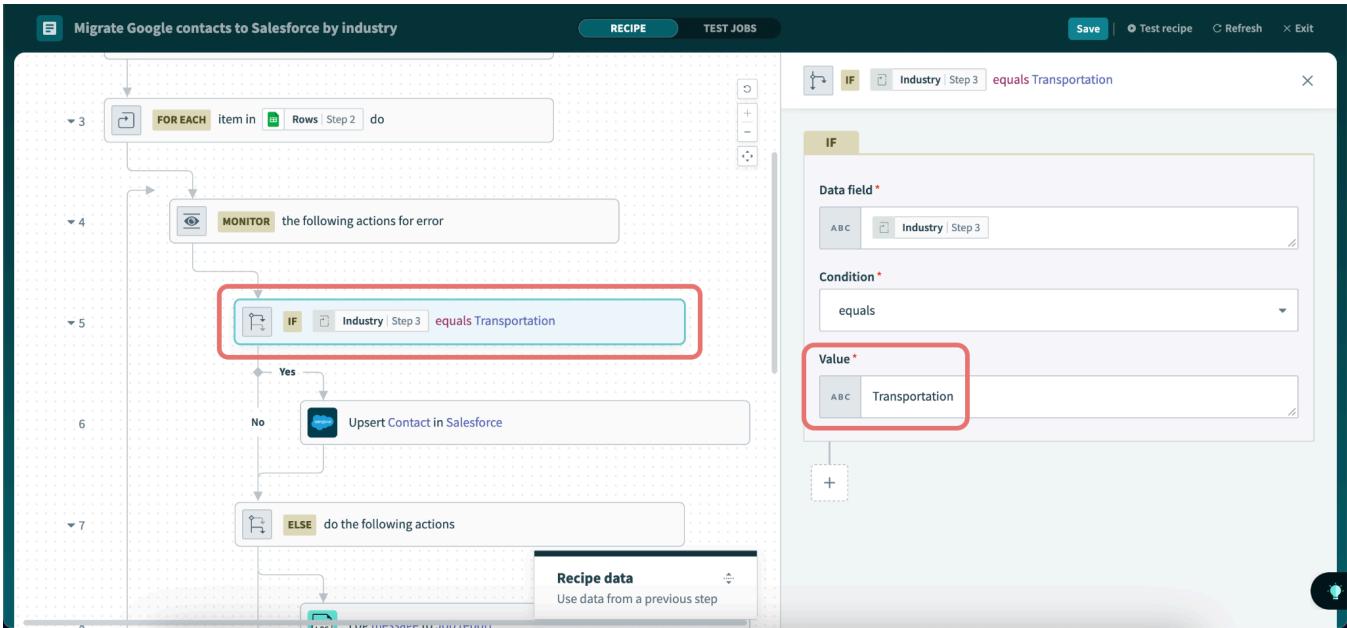
Mark the erroneous stopped job as **Failed** and map the **Error Message** datapill from the On Error Step 9 output.



## Induce error

Now that our error monitoring and handling blocks are configured to look for errors, add a bit more functionality to these blocks.

Change the Step 5 value to **Transportation** because we are done importing finance contacts.



Add a **Status** column to your Google Sheet so we can track migration status while the recipe is running.

A	B	C	D	E
First Name	Last Name	Email Address	Industry	Status
Tariq	Miah	tariq@company1.com	Finance	
Walter	Jenkins	wjenkins@_company2.com	Transportation	

Modify one of the emails slightly, as to induce an error.

Walter	Jenkins	wjenkins@_company2.com
--------	---------	------------------------

Let's **add an underscore** to the beginning of Walter's email address

## Configure successful/unsuccessful messages

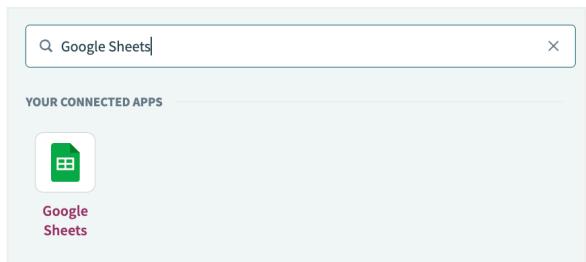
Under Step 6 to upsert a contact in Salesforce, let's add a step and add an action in an app.



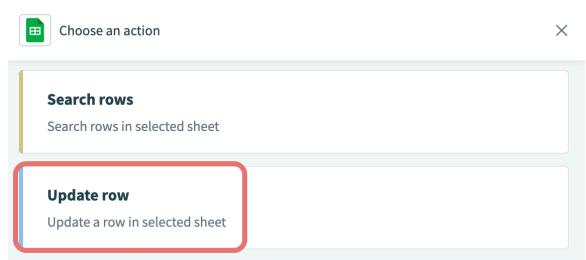
To do this, click the **plus (+) icon** and choose **Action in an app**.



Choose **Google Sheets** as the app.



Choose **Update Row** as the action.

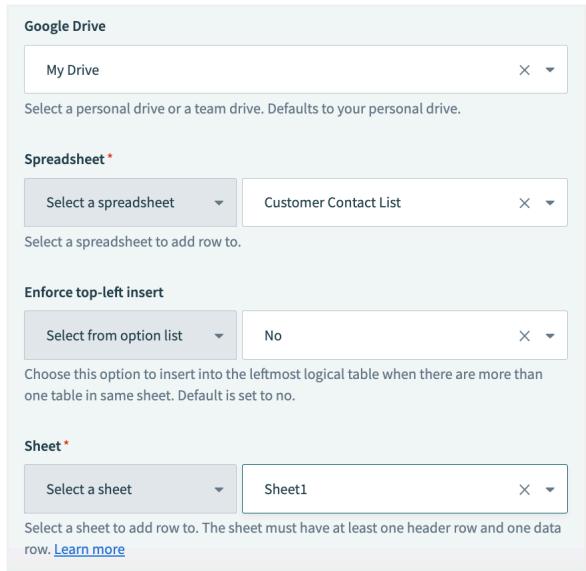


Once connected, we will finish setting up our Update Row action.

For **Google Drive**, choose your drive from the dropdown list.

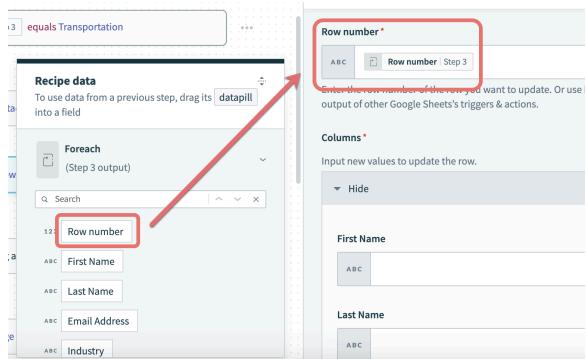
For **Spreadsheet**, choose the **Customer Contact List** spreadsheet we created during lab prep.

For **Sheet**, choose **Sheet1**.



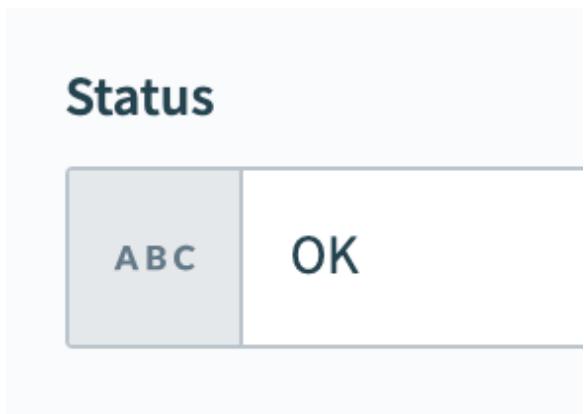
Remember, because we are working within a repeat step block, we need to map datapills from the Foreach step.

Map the Row Number datapill from Foreach Step 3 output to the required **Row Number** field.



In this step, we are logging the migration status as **OK** when the contact was successfully created in Salesforce.

In the **Status** field, type **OK**.



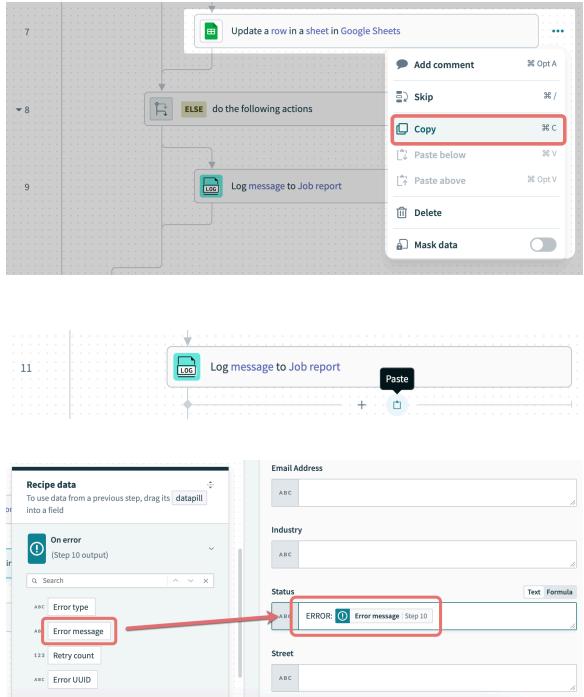
Next, we want to log any errors in our Google Sheet.

Copy the step you just created, Step 7, and paste it under **Step 11** to log message to job report.

When you copy a step, you'll see a **Paste** icon when you hover between two steps.

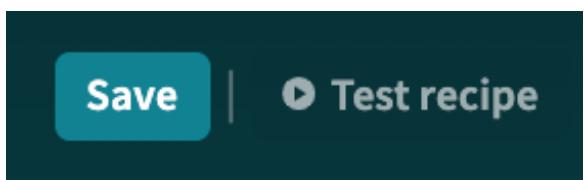
Click the **Paste** icon to automatically paste the step you previously copied.

Edit the status field to read **ERROR:** and map the **Error Message** datapill from the on error Step 9 output.



**Save and test recipe to see the induced error**

**Save** and **test** your recipe.



You'll see the recipe stopped and you'll find an error at Step 6: email is invalid.

The screenshot shows the Workato Recipe Editor interface. At the top, it displays the recipe name "Migrate Google Sheet contacts to Salesforce by industry v7" and the status "Test finished". A message at the top indicates a failure: "Failed at step 6" with the error message "Entered email id is not valid: wjenkins@\_company2.com". Below this, the recipe structure is shown with two steps: "Trigger on a specified schedule" and "Get rows from sheet in Google Sheets". To the right, a modal window for the "Upsert Contact in Salesforce" action is open, showing the input tab with the same error message.

## Verify output and test recipe again without Stop Job step

When you verify your recipe output in your Google Sheet, you'll see the **Status** column has been updated to include an error.

The screenshot shows a Google Sheets document titled "Customer Contact List". The spreadsheet has columns for First Name, Last Name, Email Address, Industry, and Status. The "Status" column contains an error message: "ERROR: Entered email id is not valid: wjenkins@\_company2.com" for the row where the email address "wjenkins@\_company2.com" was entered. The rest of the data is valid, with industries like Finance, Transportation, Retail, Healthcare, Education, and Power listed.

	A	B	C	D	E	F	G	H
1	<b>First Name</b>	<b>Last Name</b>	<b>Email Address</b>	<b>Industry</b>	<b>Status</b>			
2	Tariq	Miah	tariq@company1.com	Finance				
3	Walter	Jenkins	wjenkins@_company2.com	Transportation	ERROR: Entered email id is not valid: wjenkins@_company2.com			
4	Francine	Thibault	fran@company3.com	Retail				
5	Avinash	Jain	avinash.jain@company4.com	Healthcare				
6	Alessia	Zullo	alessia@company5.com	Education				
7	Irma	Klein	klein@company6.com	Power				

When you verify this in Salesforce, you'll notice no other records were processed and no contacts were added.

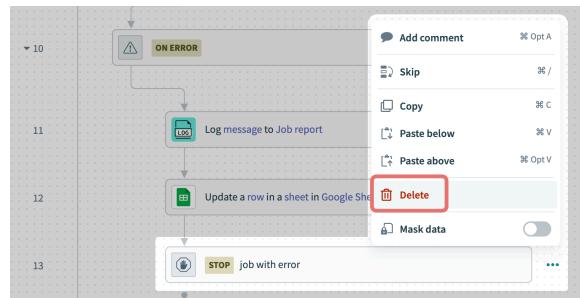
This is because the recipe was configured to stop as soon as an error occurred.

The screenshot shows the Salesforce interface with the 'Sales' tab selected. In the top navigation bar, the 'Contacts' tab is highlighted. Below it, the 'Recently Viewed' section displays four items, all updated a few seconds ago. The list includes Tariq Miah, Kelsey Sutton, Aisha Tinibu, and Clark Kent, each with their email address and a 'Training' status indicator.

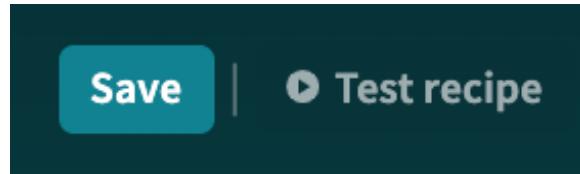
Back in Workato, expand the **Stop Job** step to see output for this step.

If you'd like your recipe to continue running after an error has occurred, simply remove the **Stop Job** step.

To do this, **delete** Step 13 to see what that looks like for this recipe.



When you **save** and **test** your recipe again, you'll see that all transportation industry customers have been processed and migrated into Salesforce, except for any contacts with invalid emails.



## Lab 4: Data modification and data mapping expanded

### Add data to Google Sheet

**Add three columns** to our Google Sheet:

- Street
- Phone
- Birthdate

Street	Phone	Birthdate

**Download** our CSV file for sample data that you can **copy and paste** into your Google Sheet.

Alternatively, you can manually fill these columns with the following values:

- In the **Street** column, add street names to each customer contact **using underscores instead of spaces** as some systems may record data this way.

Salesforce, however, expects street names formatted differently, so we'll have to clean up this data in our recipe.

- Under **Phone**, add **10-digit** numbers for each customer contact.

We'll also format this data for Salesforce.

- In the **Birthdate** column, add birthdays for each customer using the pattern of four-digit year, two-digit month, and two-digit day of the month.

Street	Phone	Birthdate
21_Spurs_Ln_245	2104877463	1987-06-21
1118_Euclid_Ave_NE	4048634129	1976-08-08
2323_Oak_St	8109855255	1987-05-15
1_Marine_Way	4158681632	1993-04-06
220_Tom_Hill_Sr_Bld	4784747582	1961-05-19
6110_Derry_St	7175619895	1959-03-16
121_Shue_Dr	9373942108	1977-12-10
205_E_Hospitality_Ln	9098887571	1987-09-27
231_Central_Fwy	9405691400	1996-10-31
709_U.S._71	4793941131	1997-09-21
7490_S_University_Bld	3032212038	1997-08-14
368_Patricia_Ave	7277363008	1968-07-03
6770_New_York_Ave	5299930901	1987-04-17
5237_34th_St_N	8778878560	1957-04-21
19801_Gulf_Fwy	2813380459	1992-07-20
1505_Dillingham_Bld	8088420288	1966-12-28
11111_Santa_Monica_Bld	9517103179	1977-03-15
86_Anderson_St	2015250885	1985-08-03
3940_E_65th_Ave	9075295088	1992-03-12

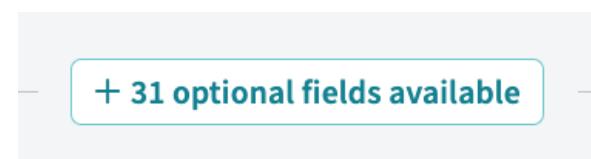
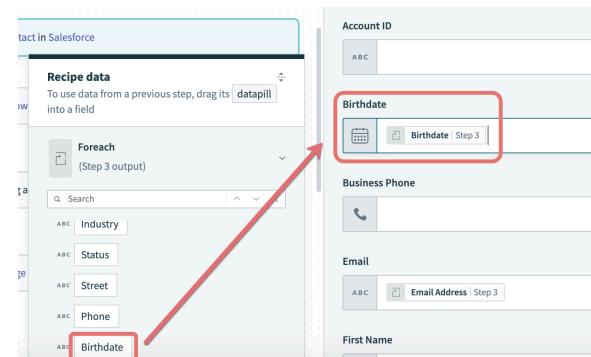
## Map and transform new data

Back in Workato, let's map and transform this additional data.

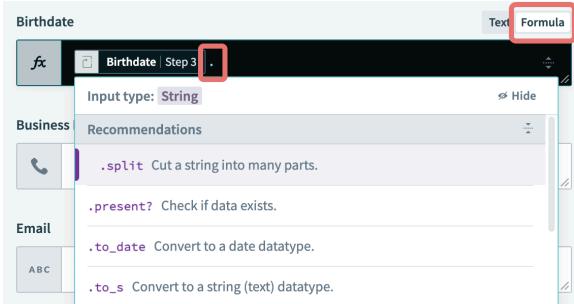
Open the recipe from our last lab and ensure you are in recipe edit mode.

In the **Upsert Contact in Salesforce** step, which should be Step 6, locate and map the Birthdate datapill from the Foreach Step 3 output.

If you are unable to locate the Birthdate datapill, **expand optional fields**.



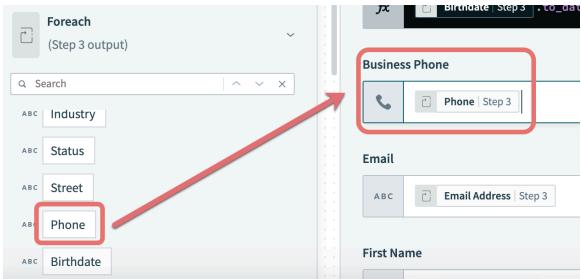
Toggle formula mode and type a **period** or **dot** to preview a list of suggested functions.



Locate or enter the function to convert the data to a string of time in the pattern of day of the month with zero prefix, month with zero prefix, and year with century.

You'll see this as the **to\_date.strftime** formula, with the day, month, and year operators as shown:  
("%d-%m-%Y")

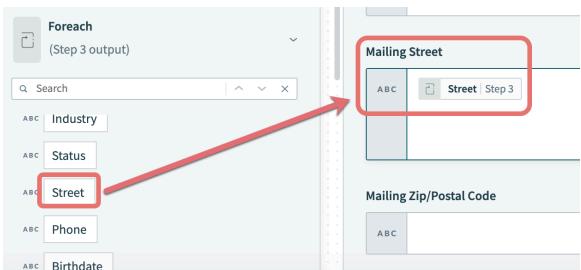
In the **Business Phone** field, let's map our **Phone Number** datapill from Step 3.



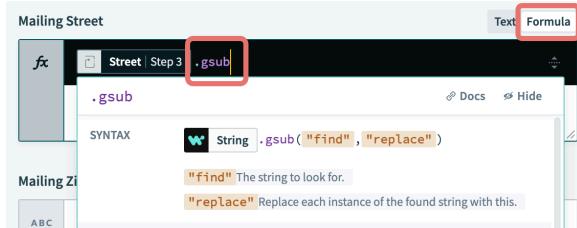
Toggle formula mode and add **.to\_phone** to the formula.



In the **Mailing Street** textbox, map the **Street** datapill from Step 3.



Toggle formula mode and add **.gsub** to the end of the formula.

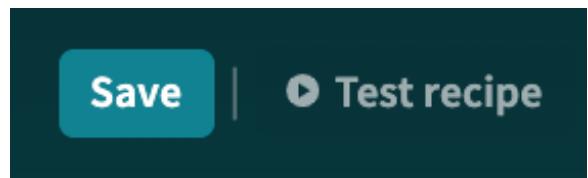


Now, let's customize the parameters of the **gsub** function to replace all underscores in the street name with spaces: **.gsub(" "," ")**

## Save and test your recipe to view results

**Save** and **test** your recipe.

After a successful job completion, view the results of your recipe edits in the Contact tab of Salesforce.



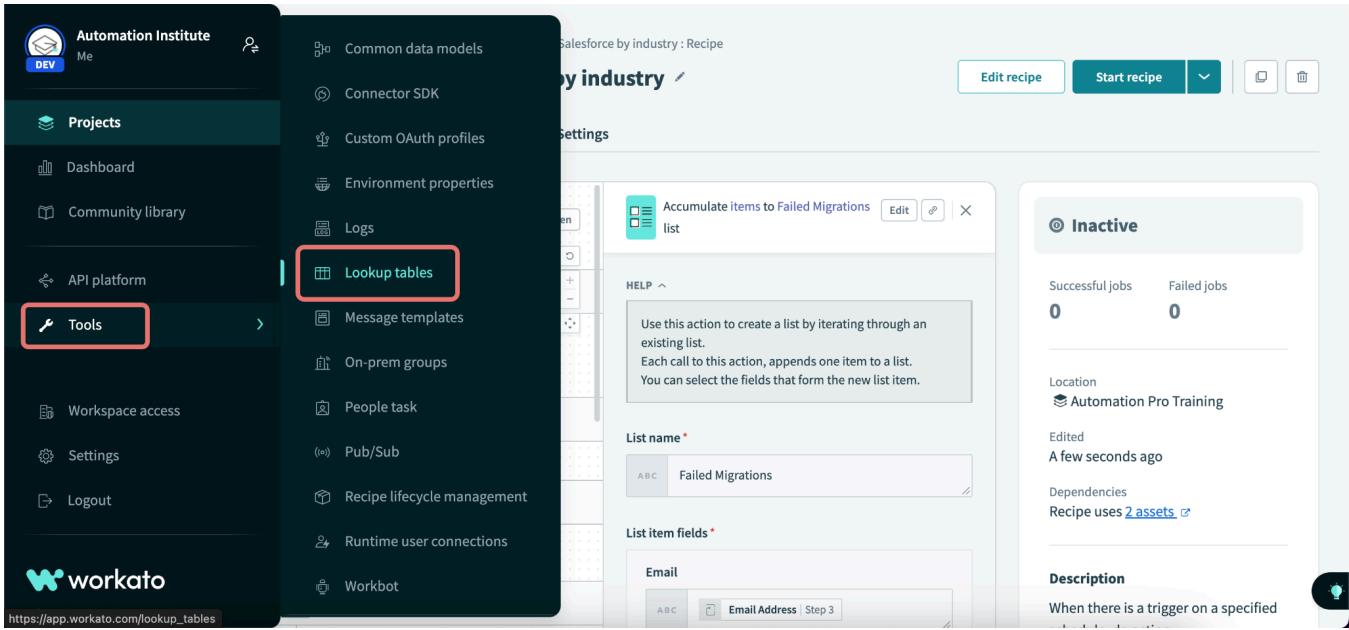
You can also view the status of your edits in your Google Sheet.

Change the industry identified in your conditional action to update data for customers in other industries.

## Lab 6: Lookup tables

### Create a lookup table

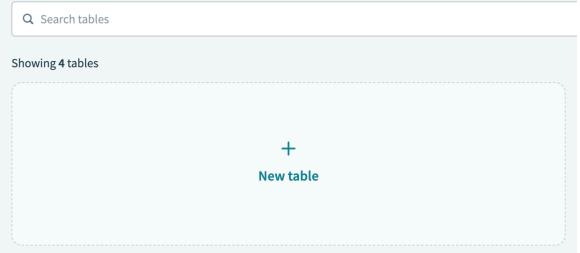
In the navigation bar along the left locate **Tools** and then **Lookup Tables**.



The screenshot shows the Workato Platform interface. The left sidebar is dark-themed and includes options like 'Dashboard', 'Community library', 'Logs', 'API platform', 'Tools' (which is highlighted with a red box), 'Workspace access', 'Settings', and 'Logout'. The main workspace is titled 'Salesforce by industry : Recipe' and contains a 'Settings' section. A specific action named 'Accumulate items to Failed Migrations' is selected. The 'List name' is set to 'Failed Migrations' and the 'List item fields' dropdown shows 'Email' is selected. To the right, there's a summary card for the 'Failed Migrations' list, indicating 0 successful and 0 failed jobs, located in the 'Automation Pro Training' workspace, edited a few seconds ago, and using 2 assets.

**Click New Table.**

### Lookup tables



The screenshot shows the 'Lookup tables' page. It features a search bar at the top labeled 'Search tables'. Below it, a message says 'Showing 4 tables'. A large rectangular area displays four tables, each represented by a small icon and some text. At the bottom right of this area is a blue '+' button with the text 'New table'.

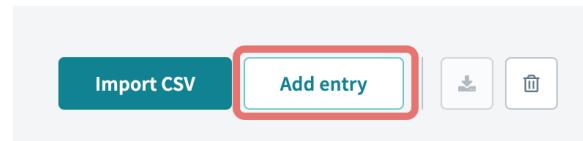
Name this table **US Postal Codes** and start adding entries manually.

The screenshot shows the workato interface with a sidebar on the left containing icons for Dev, Data, Projects, and Help. The main area is titled "Lookup tables > Untitled lookup table" and contains a table named "US Postal Codes". The table has a status bar indicating it is available in all projects, updated on 11/04/2023 at 10:49 PM, and has 0 entries. A central message says "Let's add some entries!" with instructions to make sure the first row of a CSV file contains column names and that the table can have up to 10 columns and 100,000 entries. Two buttons are present: "Import CSV" and "Add entries manually", with the latter being highlighted by a red box.

Name the first column **Postal Code** and the second column **City**.

The screenshot shows the "US Postal Codes Lab" view. The sidebar and top navigation are identical to the previous screenshot. The main area displays a table with two columns: "Postal Code" and "City". A search bar labeled "Search entries" is at the top, followed by a message "Showing 0 entries". Below the table are "Import CSV" and "Add entry" buttons, with "Add entry" being highlighted by a red box. There are also icons for download and delete.

Click **Add Entry** to start adding rows matching postal codes to their respective cities.



Enter postal code **00001** and **Workato HQ** for city.

Enter postal code **00002** and enter **Automation Institute** for city.

Showing 2 entries

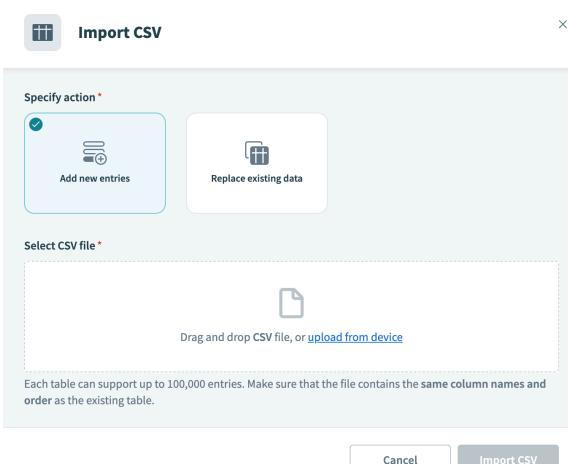
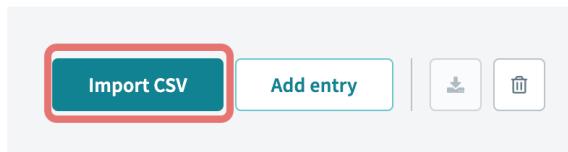
Postal Code	City	...
00002	Automation Institute	
00001	Workato	

But there's got to be a faster way to enter thousands of postal codes, right? Absolutely.

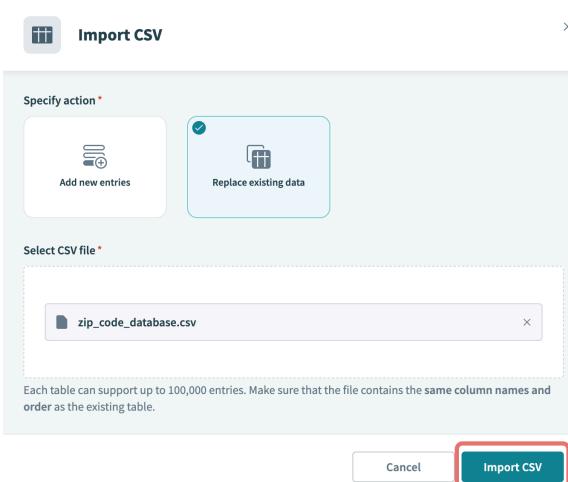
**Download** our US postal code CSV for import into your lookup table.

When importing, you can add to your table or overwrite all existing data.

If prompted, be sure to select **Do not import first row header**.



Click **Import CSV** to easily populate your US postal codes lookup table.



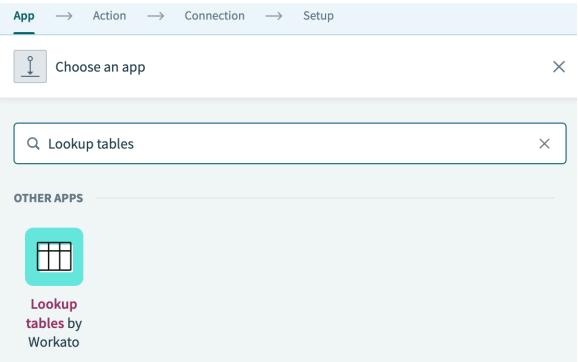
## Invoke your lookup table in a recipe

Let's edit the recipe we've been working on for the past few labs.

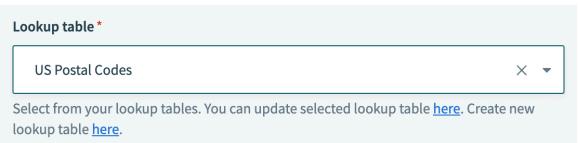
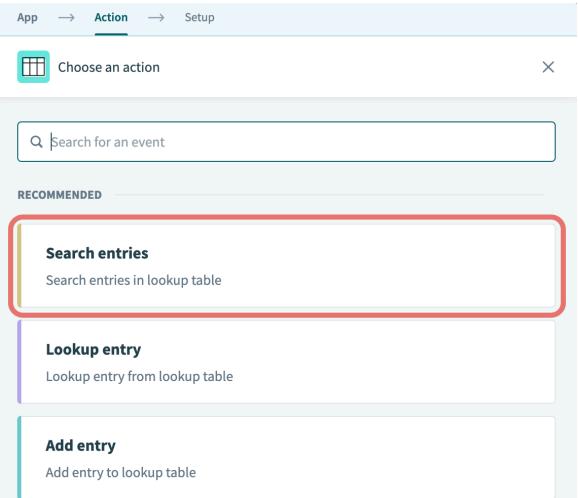
Under the Monitor step and just before the IF step, add a step to add an action in an app.

To do this, click the **plus (+) icon** and choose **Action in an app**.

For app, search for and choose **Lookup Tables by Workato**.



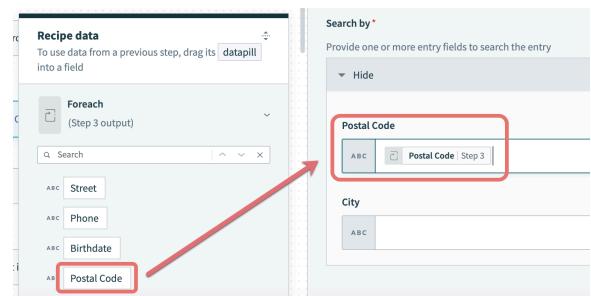
Select **Search Entries** as the action.



Select our US postal codes table as the lookup table.

In the **Search By** block, we're going to map the postal code datapill from our Foreach step to the **Postal Code** field.

This will instruct Workato to search entries by postal code provided in the source data, which is our Google Sheet.



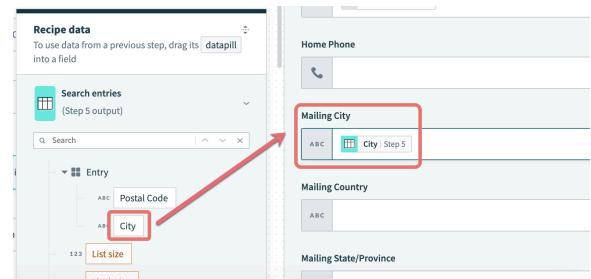
## Map and transform customer data

Click on our **Upsert Contact in Salesforce** step and locate your **Search Entries** output in recipe data.

There you will find the postal code and city objects found in your lookup table. You may have to expand the **Entries** object.

Map the **City** datapill from Step 5 output to the **Mailing City** field.

This will instruct Workato to map data from our lookup table based on the postal code that was looked up in the Search Entry step.



## Save and test your recipe

**Save** and **test** your recipe.

After a successful job completion, check Salesforce for updates to your Contacts tab.

Change the industry identified in your conditional action to update data for customers in other industries.

