Key Concepts

Quick Reference Card

WHAT ARE RECIPES?

Recipes are simple-to-build yet powerful connections between your apps that automates your work for you, enabling you to work smarter and better — with less effort. We built the Workato platform to manage the integrations between apps in the background for you so you can create your own integrations intuitively in minutes without requiring knowledge of programming.

You can create recipes from scratch or use existing recipes shared by the Workato community. A recipe contains a Trigger followed by one or more Actions. The trigger determines when your recipe will be executed and the actions determine what work your recipe will perform automatically. In a recipe you can connect multiple apps, have multiple actions and conditional actions.

A public recipe can be viewed by anyone on the web, even without signing in. However, if you have subscribed to a paid account, you'll have the option to keep your recipe private and they won't be seen by others, unless you explicitly share it.

COMPONENTS OF A RECIPE

UNIQUE ID
A unique number appended to your recipe

NAME
A name or title which can be modified

TRIGGER
One trigger, which specifies what event to look out for

ACTION(S)
One or more actions, which takes place when the trigger fires

JOB HISTORY
Will exist when the recipe has processed at least one record

START BUTTON
Starts and stops the recipe

COPY/DELETE BUTTON
Creates a copy (duplicates) the recipe or deletes it completely

SOCIAL SHARING BUTTONS
Like or share via email or Twitter

RECIPE DESCRIPTION
Lets you write a simple or elaborate description for your recipe

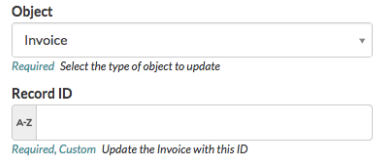
APP CONNECTION PARAMETERS
Specifies the sign-in credentials to login to each app

SAVE RECIPE BUTTON
Saves the current settings of your recipe to the cloud.

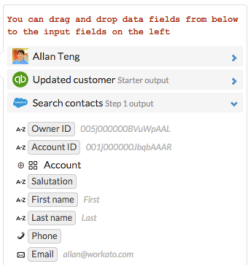
https://support.workato.com/solution/... *(with screenshots)*

GLOSSARY OF TERMS

INPUT FIELDS
Input fields are the fields that are used to map data into. Most actions have input fields and you can find them on the left side of the recipe.



DATA FIELDS / DATA PILLS
Data Fields (also known as Data Pills) are data from previous steps that you can drag and drop into inputs fields on the left. Doing so simply maps the Data Fields into the Input Fields.



TRIGGER
Each recipe has only one trigger, which describes and tells the system when to start a recipe.

ACTION(S)
Actions describes the linear sequence of work your recipe performs when it is triggered. There are 4 types of actions — *action, conditional action, repeat action, conditional stop.*

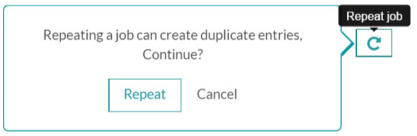
FEATURED RECIPES
Featured Recipes are high quality recipes that are widely used by the Workato community.

workato featured

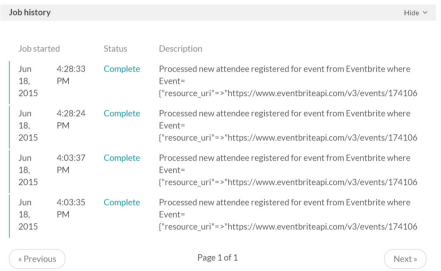
ACTIVE / INACTIVE RECIPES
While logged in, your main Workato page displays running recipes in the 'Active' tab and displays stopped recipes in the 'Inactive' tab.

Active (7) | Inactive (69)

REPEAT JOBS
You can repeat a job that was previously processed by clicking on the Repeat Job button on the right of every job (in Job History).



JOB HISTORY
Job History shows a list of completed jobs in a recipe. There are a few states — Processing, Warning, Complete, Error.



RECOMMENDED RECIPE
Recommended recipes are recipes you might find useful based on apps that you are using. You can get your own copy of a recipe by clicking 'Get recipe', and start using it to integrate your apps.

How To

SEARCH

https://support.workato.com/solution/categories/1000062615/folders/1000098653/articles/1000186445-how-do-i-search-for-recipes-

Workato allows you to search from thousands of recipes in a quick and simple way. Pull up the EXPLORE page from the top navigation bar and simply type in the relevant apps to search. For example, entering *'salesforce quickbooks invoice'* will bring up a list of recipes that you can 'Get' and use (or customize).

Search for	*Example*
apps	salesforce gmail
keyword	invoice
user	will
combinations	sheets quickbooks estimate
acronyms	sfdc, qbo
exclusion search	salesforce -clock
user exclusion search	eventbrite -user_name:samuel

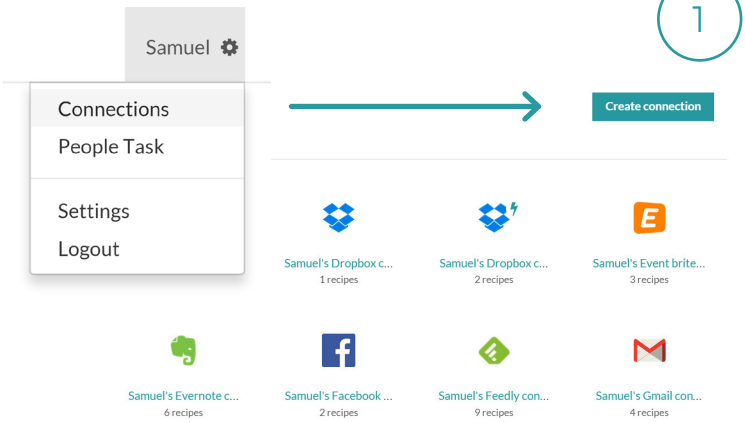
APP CONNECTIONS

https://support.workato.com/solution/categories/1000062615/folders/1000098653/articles/1000167157-app-connections-how-do-i-connect-to-apps-why-do-i-need-it-

Most apps require that you are connected to them before you can exchange information with them. Notable exceptions are the Clock and Email app. You may not see all the information for the recipes if you are not connected — stuff like trigger events, sample data for the input fields on the right.

There are two ways to connect with your apps.

1



From the top right of the page, access your button with your name and head to the 'Connections' page. Then, create a new connection using 'Create connection'.

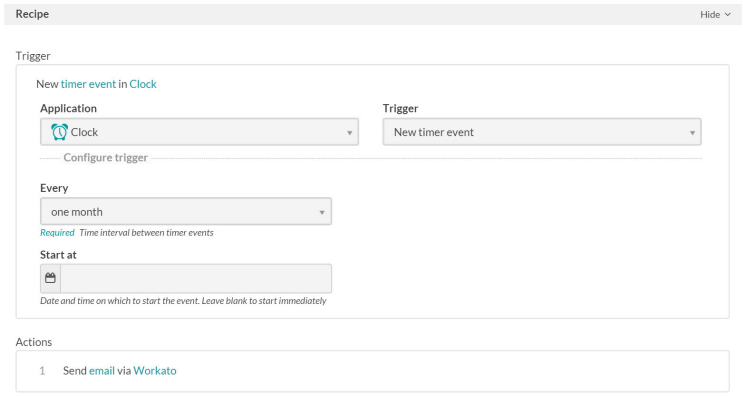
The other way is to connect them within the recipe page itself in either Step 3 when creating a new recipe, or at the bottom of the page under the 'Connections' section.

2

TRIGGER AND ACTIONS

https://support.workato.com/solution/categories/1000062615/folders/1000098653/articles/1000166851-triggers-and-actions-in-workato

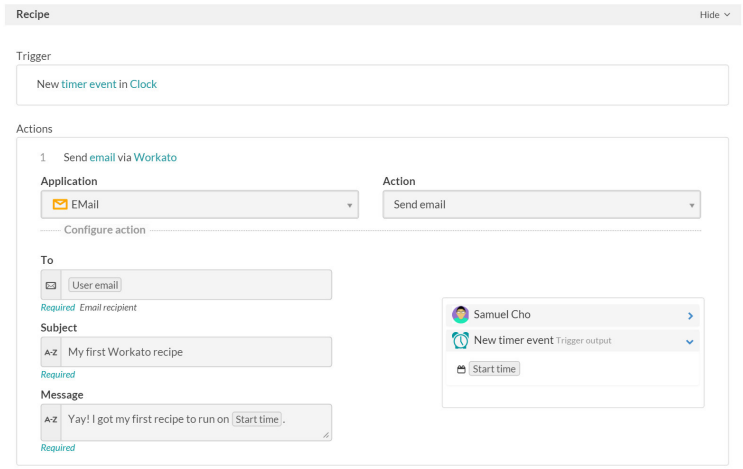
Triggers are and Actions are the key components of a recipe.



The Trigger defines the event that this recipe is watching for. In this example it is looking for a new timer event by the Clock app in Workato. Every trigger has at least these two elements:

1 The App (in this case, Clock) and
2 The Event (in this case, 'New Timer event').

Some triggers have additional parameters to specify how far back it will go to look for the events. For example for Salesforce, you can specify that you want to pick up 'New leads' from a week back or from a day back, and so on.



The Trigger is followed by the Actions. The actions specify what the recipe should do when the trigger fires. When a 'New Timer event' is started in clock, grab the time when the recipe started and send it to the user with the email body specified.

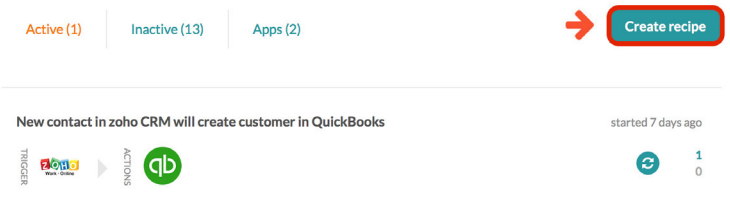
There are other types of actions; four types in fact — *action, conditional action, repeat action* and *conditional stop*. Details for the conditional actions and the repeat action are explained later.

CREATING / COPYING A RECIPE

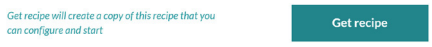
https://support.workato.com/solution/categories/1000062615/folders/1000098653/articles/1000165710-how-do-i-create-or-clone-a-recipe-

You can view a recipe as a guest, but to create, change or copy recipes you must be logged into Workato.

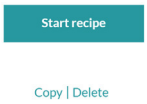
You can create a recipe from scratch very easily. On the home page (accessible by clicking on the top left Workato logo), click on the "Create Recipe" button.



To get a copy of someone else's recipe, click 'Get Recipe'.



To make a copy of your own recipe, click 'Copy' underneath the 'Start recipe' button.

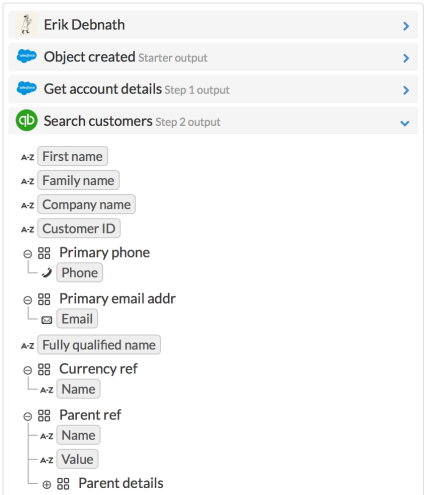


INPUT FIELDS

https://support.workato.com/support/solutions/articles/1000166846-configuring-actions-understanding-the-input-fields

For actions such as to create or update or add line items, they need to be configured — which means you have to provide input data for each action. To do this, you are presented with two things.

On the left, there are blank fields that need an input. Values can be entered in multiple ways. You can enter static text/numbers (known as hardcoding), select from dropdowns, map data from available data fields or input certain formulas (covered later).



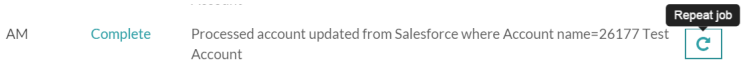
On the right, we have all the fields available to you, the Data tree. These are records you have retrieved or created so far. The first one is the user details of the logged in user, and in the example below the next ones are Salesforce custom object details that was from Create Object action, Salesforce account details from the Get Account details action, and QuickBooks customer details from Search customers action. The rounded gray boxes are called Data pills and come in a variety of data types.

List data is unique because it contains multiple lines of data — putting that in a input field would only result in the use of the first line of data.

See later article — *Working with lists.*

REPEAT JOB

https://support.workato.com/support/solutions/articles/1000153137-how-do-i-repeat-a-job-should-i-rerun-or-repeat-jobs-will-it-create-duplicates-
https://support.workato.com/solution/categories/1000062615/folders/1000200900/articles/1000144426-rerun-a-failed-job-how-and-when-should-i-rerun-a-job-



Any job in the job history overview, regardless of if it completed or failed, can be rerun by clicking on the green arrow icon on the rightmost side of each job. There are two options:

OPTION 1

If you run the job while the recipe is running it uses the *original (old)* recipe definition (when it was first run).

OPTION 2

If you stop the job and then rerun, then the *new* recipe definition will be used.

NOTE (1)

Duplicates may occur when jobs are rerun, so be sure to check and remove duplicates (if present) before re-running the job.

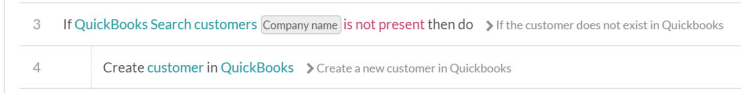
NOTE (2)

When you are testing recipes or re-running a failed job, always stop the recipe first before repeating the job to ensure it uses the new schematics/instructions/definition when processing the data.

NESTED STEPS

https://support.workato.com/support/solutions/articles/1000142722-how-can-i-implement-nested-steps-in-a-recipe-

A nested step is denoted by the slight indentation as seen for Step 4:



All steps indented under a "for" loop will only be executed within the loop, and all steps indented under a conditional "if" step will only be executed if the conditional step is true. If the condition is not met, the recipe does nothing.

To add multiple nested steps within a "for" loop or a conditional "if" step, select the up and down arrows on the nested step and choose the step you want to add.

To get multiple doubly nested steps, again just select the up and down arrows on the nested steps and choose the step you want to add. You can nest steps almost indefinitely.

MISSING CUSTOM FIELDS

https://support.workato.com/support/solutions/articles/1000172114-i-can-t-see-all-the-fields-to-map-why-are-the-fields-missing-for-the-action-

When you open up an action, only the most important fields that are to be mapped show up. This is to reduce clutter on the page. A 'Show more' link is located at the bottom.

You can do the same thing with the data tree on the right, where the same 'Show more' and 'Show less' buttons will show up.

REFRESH(ING THE API)

If you have added a new custom field into your Salesforce account while editing the recipe, you will notice that it does not show up. You need to manually refresh the step by going to the 'Action' dropdown and change the action to another, and then back again.

CONDITIONAL STEPS

https://support.workato.com/support/solutions/articles/1000153148-what-are-conditional-actions-and-what-conditions-do-they-support-

There are four types of actions — *action, conditional action, repeat action* and *conditional stop*. Two of these actions, Number (2) and (4), are conditional.

Conditional action (2) will perform the actions that are under it only if the condition is true. For example, if the account is not present in Salesforce, then create an account and send out an email.

Conditional stop (4) will stop processing the recipe if the condition is true. You **cannot** have any other actions nested under it. For example, if the account is present in Salesforce, then stop.

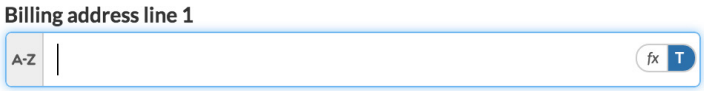
Conditional actions support 11 conditions. In general, all the conditions except "is present" and "is not present" checks for the value of the specified string. On the other hand, "is present" and "is not present" conditions check if there is a value present or not.

CONTAINS	True if the specified field contains the specified string (word/letters/numbers). False otherwise. Case sensitive.
STARTS WITH	True if the specified field starts with the specified string (word/letters/numbers). False otherwise.
ENDS WITH	True if the specified field ends with the specified string (word/letters/numbers). False otherwise.
EQUALS TO	True if the specified field exactly matches the specified string (word/letters/numbers). False otherwise.
NOT EQUALS TO	True if the specified field does not match the specified string (word/letters/numbers). False otherwise.
GREATER THAN	True if the specified field is greater than the specified object. False otherwise. This can be used to compare numbers, dates and text.
LESS THAN	True if the specified field is greater than the specified object. False otherwise. This can be used to compare numbers, dates and text.
IS TRUE	Checks for boolean (true/false) values.
IS NOT TRUE	Checks for boolean (true/false) values.
IS PRESENT	True if field is not null. False otherwise.
IS NOT PRESENT	True if field is null. False otherwise.

Formulas

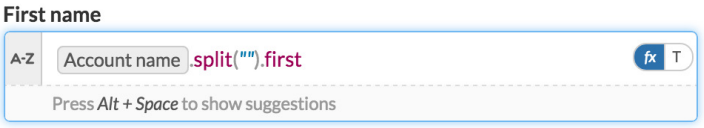
HOW TO USE

When you are defining the input for a field, you can be in text mode (T) or formula mode (*fx*), indicated by the box at the right of the field. You can toggle between the two modes.



The default mode is text mode. In text mode you can assign plain text or map any field from the input data you have. You can add more than one field, for example, you can put the First name and Last name next to each other. Spaces and symbols work here.

When you toggle to formula mode, you see a slightly different view of the input field. It is easily identified by the presence of the extra block below the field that says 'Press Alt+Space to show suggestions.



Formula mode accepts operators and methods to transform the data in the input field. You can bring up a simple list of commonly used Ruby methods by appending a period to the end of your data or by pressing Alt+Space. The choices will dynamically show up below, and is dependent on the type of the data - number, text, date etc.

ARITHMETIC OPERATORS

Workato supports one of the key basic arithmetic operations. If *Amount* is 45 and *Qty* is 5, the results will be as follows:

+	Addition: Adds values on either side of the operator.	Amount + Qty = 50
-	Subtraction: Subtracts right hand operand from left hand operand	Amount - Qty = 40
*	Multiplication: Multiplies values on either side of the operator	Amount * Qty = 225
/	Division: Divides left hand operand by right hand operand	Amount / Qty = 9
%	Modulus: Divides left hand operand by right hand operand and returns remainder	Amount % Qty = 0

OTHER OPERATORS

https://support.workato.com/solution/articles/1000173152-how-do-i-use-formula-mode-

Operator	Explanation (Scenario)	Examples	Output
ago	Goes back a specified period of time, with the base time as server time when job occurs.	**Taking date of job as 6/15/2015:** 1.days.ago 1.months.ago **If [pill] is 3:** [pill].days.ago [pill].months.ago	**Taking date of job as 6/15/2015:** 5/14/2015 5/15/2015 **If [pill] is 3:** 6/12/2015 3/15/2015
blank?	Checks to see if [pill] is empty. If pill is empty, returns true. If pill is not empty, returns false. Null values and blank spaces count as empty.	**If [pill] is null:** [pill].blank? **If [pill] has an empty string " ":** [pill].blank? **If [pill] has a non empty string " x":** [pill].blank?	**If [pill] is null:** true **If [pill] has an empty string " ":** true **If [pill] has a non empty string " x":** false
capitalize	Capitalizes the first letter of the entire string only	"Double Bubble".capitalize "Double. Bubble".capitalize	"Double bubble" "Double. bubble"
casecmp	Compares 2 strings, and disregards case sensitivity. Returns 0 if strings are identical, returns 1 if [pill] has a greater ASCII value, and returns -1 if [pill] has s smaller ASCII value.	**If [pill] is the string "Double bubble":** [pill].casecmp("DOUBLE BUBBLE") [pill].casecmp("DOUBLE UBBLE") [pill].casecmp("DOUBLE BUBBLE BUBBLE")	**If [pill] is the string "Double bubble":** 0 1 -1
days	Units of measurement for dates. Can be used to add or subtract days from a date.	**If [pill] is the date 6/15/2015:** [pill] + 1.days [pill] + 1.months [pill] - 12.months	**If [pill] is the date 6/15/2015:** 6/16/2015 7/15/2015 6/15/2014
downcase	Makes all letters in the string to be in the lowercase.	"Double bubble".downcase "DOUBLE bubble".downcase	"double bubble" "double bubble"
ends_with?	Checks to see if a string ends with a certain string. This is case sensitive. If it does, returns true. If it doesn't, returns false.	**If [pill] is "Double bubble":** [pill].ends_with?("bubble") [pill].ends_with?("Bubble") [pill].ends_with?("ubble")	**If [pill] is "Double bubble":** true false false
first	Returns the first item in a list. Can also be used to return the first n items in a list, as a list.	**If [list] = [item1, item2, item3, item4, item5]:** [list].first [list].first[2] [list].first[0..-2]	**If [list] = [item1, item2, item3, item4, item5]:** item1 [item1, item2] [item1, item2, item3, item4]
gsub	Replaces a specific letter in a string with another one.	'Jean Marie'.gsub('J', 'M') 'Jean Marie'.gsub('e', 'i')	"Mean Marie" "Jian Marii"
join	Joins elements in a list by a specified character to form a string. If no character is defined, by default, elements will just be joined together.	**If [list] = [1, 2, 3, 4, 5]:** [list].join [list].join(" ") [1, 2, 3, 4, 5].join(", ")	**If [list] = [1, 2, 3, 4, 5]:** "12345" "1 2 3 4 5" "1, 2, 3, 4, 5"
last	Returns the last item in a list. Can also be used to return the last n items in a list, as a list.	**If [list] = [item1, item2, item3, item4, item5]:** [list].last [list].last[2] [list].last[0..-2]	**If [list] = [item1, item2, item3, item4, item5]:** item5 [item4, item5] [item2, item3, item4, item5]
length	Returns the number of elements in a list.	**If [list] = [item1, item2, item3, item4, item5]:** [list].length **If [list] = [ ]:** [list].length	**If [list] = [item1, item2, item3, item4, item5]:** 5 **If [list] = [ ]:** 0
lstrip	Removes only leading white space in a string.	" Double bubble ".strip "Double bubble".strip	"Double bubble" "Double bubble"
present?	Checks to see if [pill] is empty. If [pill] is not empty, returns true. If [pill] is empty, returns false. Null values and blank spaces count as empty.	**If [pill] is null:** [pill].present? **If [pill] has an empty string " ":** [pill].present? **If [pill] has a non empty string " x":** [pill].present?	**If [pill] is null:** false **If [pill] has an empty string " ":** false **If [pill] has a non empty string " x":** true

presence	Checks to see if [pill] is null. If [pill] is null, returns null. If [pill] is not null, returns [pill].	**If [pill] is null:** [pill].presence **If [pill] has an empty string " ":** [pill].presence **If [pill] has a non empty string " x":** [pill].presence	**If [pill] is null:** null **If [pill] has an empty string " ":** " " **If [pill] has a non empty string " x":** " x"
reverse	Reverses the order of lists or strings.	**If [list] = [item1, item2, item3, item4, item5]:** [list].reverse "Double bubble".reverse	**If [list] = [item1, item2, item3, item4, item5]:** [item5, item4, item3, item2, item1] "elbbub elbuoD"
rstrip	Removes only trailing white space in a string.	" Double bubble ".strip "Double bubble".strip	" Double bubble" "Double bubble"
size	Returns the number of characters in a string if used on strings, inclusive of white spaces.	"Double bubble".size "12345 6789".size	13 10
slice	Returns a partial segment of a string. Pass in 2 parameters — the first parameter is the index that decides which part of the string to start returning from (first letter being 0 and subsequently progressing incrementally), the second parameter decides how many characters to return. If only the first parameter is passed in, only 1 character will be returned.	"Double bubble".slice(0) "Double bubble".slice(0, 6) "Double bubble".slice(7) "Double bubble".slice(7, 6)	"D" "Double" "b" "bubble"
split	Splits up a string based on certain characters. Character is case sensitive. If no character is defined, by default, strings are split up by white spaces. Null values cannot be split. An error will be thrown.	"Double bubble".split "Double bubble".split("b") "Double bubble".split("d") "Double bubble".split("D")	["Double", "bubble"] ["Dou", "le ", "", "u", "", "le"] "Double bubble" ["", "ouble bubble"]
starts_with?	Checks to see if a string starts with a certain string. This is case sensitive. If it does, returns true. If it doesn't, returns false.	**If [pill] is "Double bubble":** [pill].starts_with?("Double") [pill].starts_with?("Do") [pill].starts_with?("double")	**If [pill] is "Double bubble":** true true false
strip	Removes leading and trailing white space in a string.	" Double bubble ".strip "Double bubble".strip	"Double bubble" "Double bubble"
titleize	Converts the first letter of each word in a string to uppercase.	"double bubble".titleize "DOUBLE BUBBLE".titleize	"Double Bubble" "Double Bubble"
to_f	Changes the type of a variable to a float.	**If [pill] is "Double bubble":** [pill].to_f "Double123".to_f "123".to_f "123bubble321".to_f 123.555.to_f	**If [pill] is "Double bubble":** 0 0 123.0 123.0 123.555
to_i	Changes the type of a variable to an integer. Converts strings into the value 0.	**If [pill] is "Double bubble":** [pill].to_i "Double123".to_i "123".to_i "123bubble321".to_i	**If [pill] is "Double bubble":** 0 0 123 123
to_s	Changes the type of a variable to a string.	**If [pill] is "Double bubble":** [pill].to_s 123.to_s	**If [pill] is "Double bubble":** "Double bubble" "123"
upcase	Makes all letters in the string to be in the uppercase.	**If [pill] is "Double bubble":** [pill].upcase "DOUBLE bubble".upcase	**If [pill] is "Double bubble":** "DOUBLE BUBBLE" "DOUBLE BUBBLE"

Common Scenarios

SPLIT, JOIN; WORKING WITH ADDRESSES

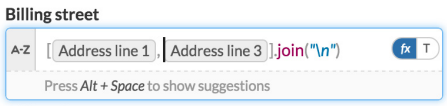
https://support.workato.com/support/solutions/articles/1000172077-how-do-i-add-new-line-in-my-formula-how-do-i-format-the-address-into-multiple-lines-

When trying to transfer text from one system to another, you may want to concatenate multiple elements or you may want to format it onto multiple lines. A good example of this is Address lines — in Xero, address lines are split into address line 1, address line 2, address line 3 but in Salesforce there is just one field, Billing street.

HOW TO DO IT

For Billing Street, turn formula mode (*fx*) instead of the default 'T'

- add a "["
- select the first field
- add a ", "
- select the second field
- close the block with "]"
- add a period "."
- key in or select join
- key in ("\n")



The basic split breaks a string into its components. It is not really useful when used on its own.
"Joseph Robinette Biden".split
returns *["Joseph", "Robinette", "Biden"]*

In *"Joseph Joe Robinette Biden"*,
"Joseph Joe Robinette Biden".split.**first** returns *"Joseph"* while *"Joseph Joe Robinette Biden".split*.**last** returns *"Biden"*.

ADVANCED MANIPULATION OF NAMES/CHARACTERS

For more advanced control, you can do fine-grained *split* and *join* them with some other character(s).

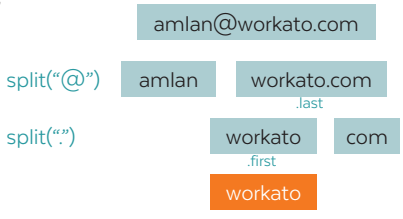
The numbering scheme for split is as follows: the array starts with 0, and ends with -1.
You specify the start and stop positions for the split.

Joseph Joe Robinette Biden
0 1 2 -1

Formula	Result
"Joseph Robinette Biden".split[0..1].join(" ")	**"Joseph Robinette"**
"Joseph Joe Robinette Biden".split[2..-1].join("@")	**"Robinette@Biden"**
"Joseph Robinette Biden".split[2..-1].join(" ")	**"Biden"**
"Joseph Joe Robinette Biden".split[2..-1].join(" ")	**"Robinette Biden"**
"Joseph Joe Robinette Biden".split[2..-1].join(".") + "@.gmail.com"	**"Robinette.Biden@.gmail.com"**

EXTRACTING DOMAIN NAME FROM EMAILS

"amlan@workato.com".split("@").last.split(".").first
returns *"workato"*



PRESENT?, PRESENCE AND NULL DATA

https://support.workato.com/support/solutions/articles/1000177311-formula-mode-present-and-presence

You can use the two functions - presence and present? to check for presence of a value and perform functions accordingly.

.present? checks to see if a value exists, and if it is present it will return one value; if not present it will return the second value.

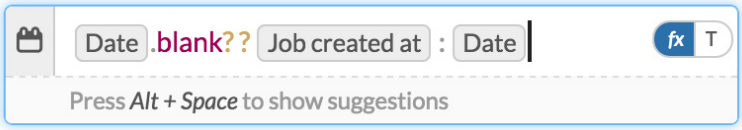
Formula	Result
's'.present? ? "a" : "b"	**"a"**
''.present? ? "a" : "b"	**"b"**
1.present? ? "a" : "b"	**"a"**
0.present? ? 0 : 1	**"0"**
[Full Name].present? ? [Full Name] : "No name given"	**[Full Name]** *if it was present*

.presence just checks to see if the value exists and returns the value only if it exists, otherwise it returns nothing.

Formula	Result
's'.presence	**"s"**
''.presence	*Nothing*

You can also use ".blank?" to check for null data, similar to ".present?"

Invoice Date



If the date is blank, it will use the date when the job was created, otherwise it will use the original date.

DATE FUNCTIONS

https://support.workato.com/support/solutions/articles/1000177317-formula-mode-date-functions

Dates are quite common elements in data structures. Occasionally, the date may not be in the right format, e.g. it may be sent as a string (which generates errors!).

".to_date" changes a string to a proper date and you may also specify the format to be converted to.

Some examples to manipulate dates:

Formula	Result
'20150420'.to_date	**Mon, 20 Apr 2015**
'04/20/2015'.to_date(format: 'MM/DD/YYYY')	**Mon, 20 Apr 2015**
'20150420'.to_date + 2.days	**Wed, 22 Apr 2015**
'20150420'.to_date - 2.weeks	**Mon, 06 Apr 2015**
'20150420'.to_date + 2.years	**Thu, 20 Apr 2017**

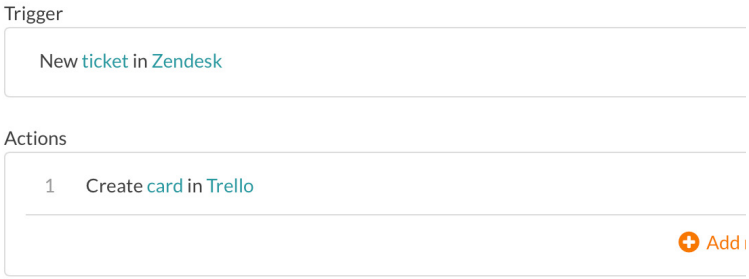
LOOKUP TABLES

https://support.workato.com/support/solutions/articles/1000185431-how-do-i-define-the-mapping-of-one-set-of-values-to-another-set-across-systems-

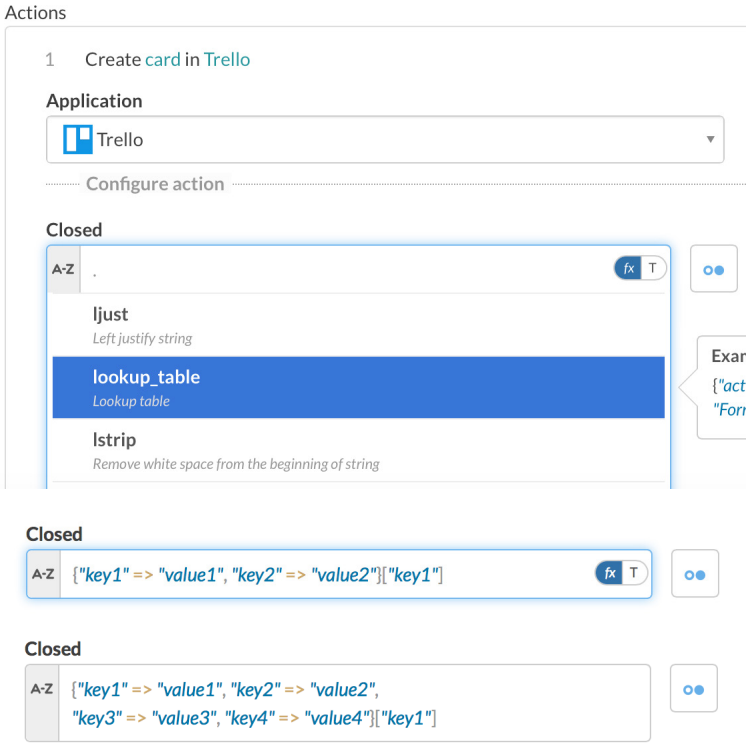
When building recipes, one may chance upon sets of fields which are conceptually similar, but their actual values don't match up.

What if I wish to sync tickets from an app with statuses such as "new, pending, solved, closed" while my project management app represents projects as only either "closed" or "not closed"?

With Workato's lookup tables, you can define which value on system A should be mapped to which value in system B. If I'm building a recipe that syncs Zendesk tickets to Trello cards:



When creating a card or updating a card in Trello upon new or updated tickets in Zendesk, I want to ensure that the Trello card's "closed" status corresponds to the Zendesk ticket status. Therefore, let's use formula mode for the input field "Closed" in Trello's create card action (or update card action), and select lookup_table.



The default example has only 2 values, so in this case you would need to enter a few more pairs.

Now, you need to replace the keys on the left-hand-side with the value of the Zendesk statuses (new, pending, solved, or closed), and replace the values on the right-hand-side with true or false, as "Closed" is a boolean field.

> *In simpler terms, Open and Closed is a true-false (yes-no) data type, which is incidentally what Trello accepts. Therefore, you need to manually specify what the tickets are converted to; i.e. a manual, one-time interpretation for the system.*

You also need to put the data pill/pillbox within the square brackets on the extreme right.

Essentially, we're saying: take the value of my Zendesk ticket, the [Status] pill, and depending on what its actual value (new, pending, solved, closed) is, substitute it out for the right-hand-side value (true or false). We then pass this right-hand-side value into the action.

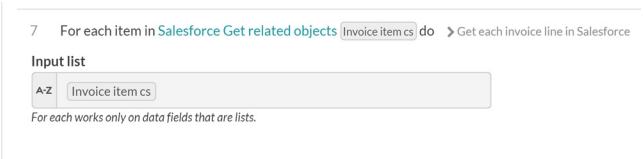
FURTHER NOTES

You can substitute the above into any other scenarios. For example, when syncing the type of employee from your HR management system into your payroll system (i.e. full-time, part-time, contractor) or for syncing the type of customer from your sales/customer management system into your accounting/customer service system (i.e. VIP, levels of priority, SLA type).

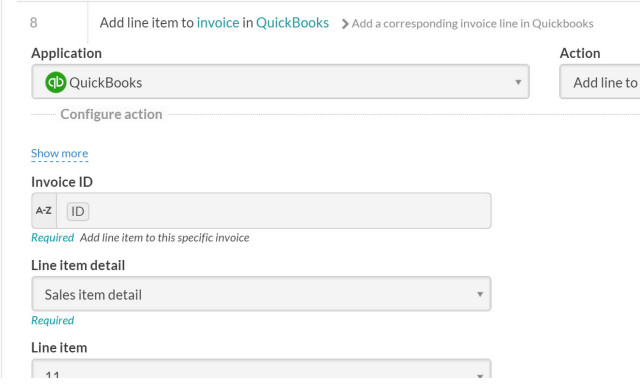
Whenever there are two sets of conceptually similar values across systems that does not map directly to each other in terms of actual labels/values, lookup tables can come in handy to map them.

WORKING WITH LIST DATA

When you encounter list data in your data tree, the normal method of mapping may not work as intended. List data is indicated by ≡. The method to use here is the action: *foreach*.



A foreach step (Step 7 in this example) has a field for the list data in which each item in the list will be processed and mapped into the other app. Place the list data pill in this field and define the actions in the following nested step (Step 8 in this example).



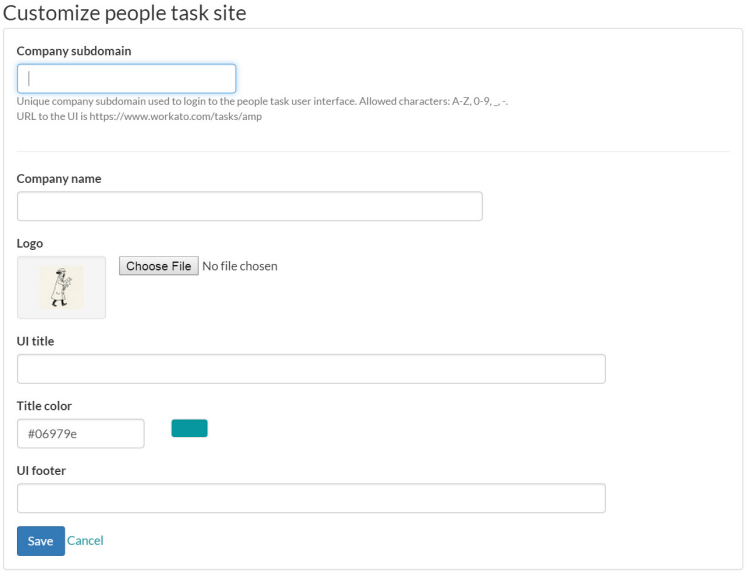
The most common use for this is to transfer line items to an invoice in an accounting app, like QuickBooks Online.
Contact us for help on more advanced manipulations of list data.

PEOPLE TASK

Workato has a powerful built-in app called People Task which allows what would be called human interaction & approval in an automated workflow process. This allows for multiple people (possibly from different departments) to coordinate within an automated recipe.

SETTING UP

Customize people task site



Access the People Task page from your profile at the top right of the page. Set up a site so that you are able to add participants.

ADDING PARTICIPANTS

People task participants

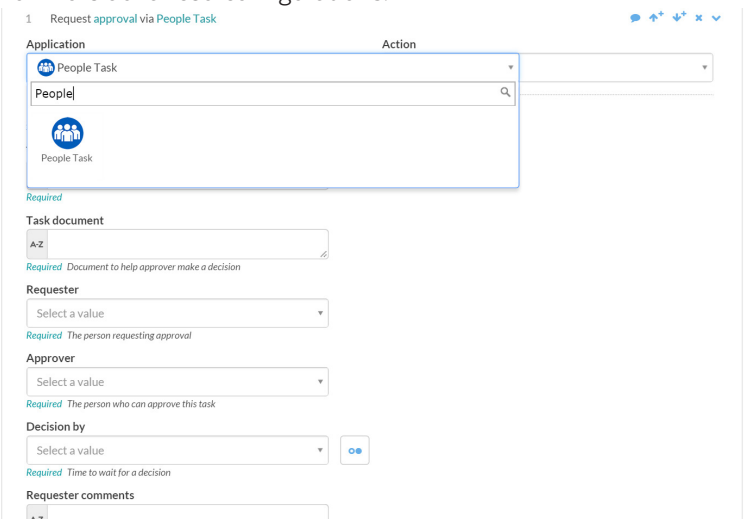
+ Add participant

Name	Email	Active
John Smith	johns@domain.com	false

Participants need to be manually added, and then verified from the email address that it was added.

USING PEOPLE TASK IN YOUR RECIPE

Add a step for People Task where approval is required. Contact us for more advanced configurations.



CLOCK

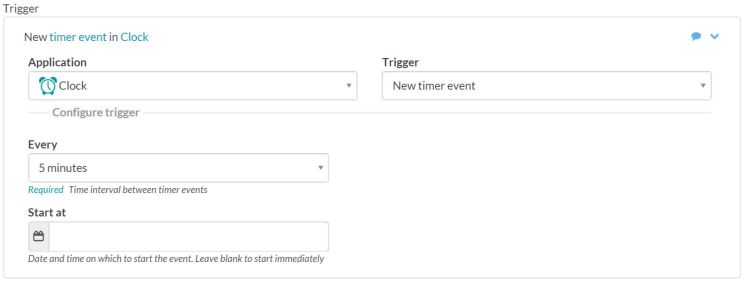
Another simple, basic yet powerful app in Workato is the clock. The clock is great for testing recipes and scheduling certain events in recipes. The clock has different functions when used in the Trigger and Actions.

TRIGGER

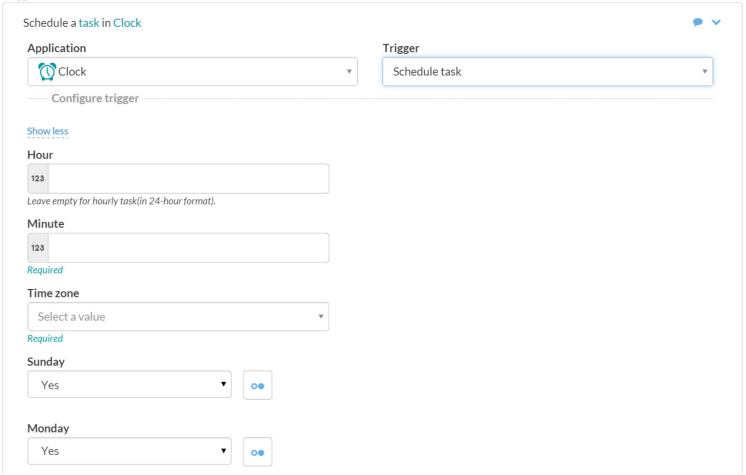
As a trigger, the Clock app can be made to trigger on a new timer event or scheduled task. The 'New timer event' is a simple repeating trigger that has a fixed time interval, e.g. 5 minutes. The 'Schedule task' function can be configured in a more precise way.

Note that in both cases, the first instance will be run immediately when 'Start recipe' is pressed.

Trigger

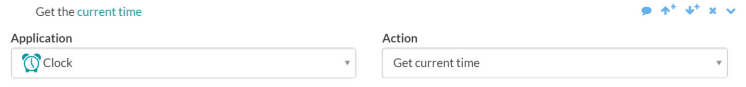


Trigger



ACTIONS

When used as an action, the Clock app can do two things; it can obtain the current time and wait for a specified time before the next action is run. This allows one to work with time within recipes.



TESTING RECIPES

To test whether an action will work when triggered in the Trigger step, you may just set your trigger to a new timer event and configure the actions in the recipe to be performed. This is especially useful if you are testing if data can be pulled from your app (visible in each job's *Input* and *Output* tab).

workato

Workato Inc
1601 S De Anza Blvd Suite 270
Cupertino, CA 95014
workato.com