



How to use Fetch actions in Zoho Flow

Business processes come in all shapes and sizes. Your automation workflows need to adapt to these processes for maximum productivity. Sometimes when you build these flows, you might not find all the data you need. This is where fetch actions can help. They look for details of a record and bring them to your flow. You can now set up your flow to use this data contextually and perform other tasks.

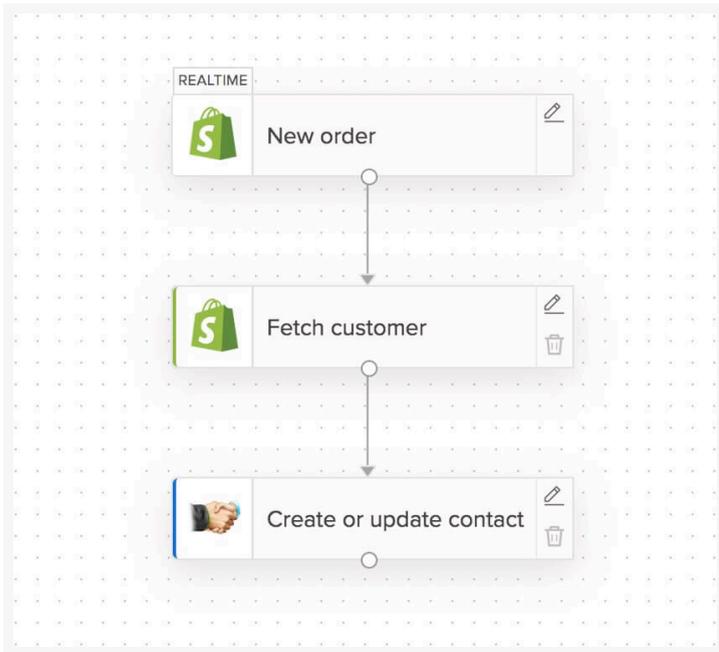
Fetch actions are app-specific and can be found alongside other actions of an app. Depending on what the app supports, fetch actions can get more details based on a value such as the record name, email, or entity ID (e.g.: Lead ID for leads). In this article, we'll discuss how to customize your flows using the fetch actions.

Here are a few scenarios where fetch actions can help:

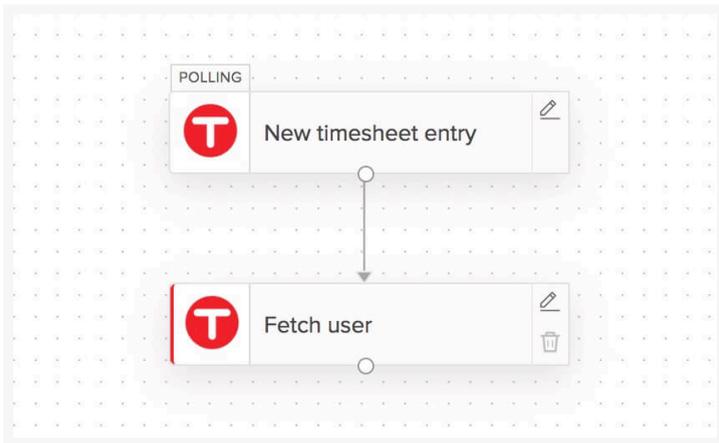
When the trigger or action doesn't provide enough data

A trigger or an action might not provide data that you want to use in the next steps of your flow. Fetch actions bring this data, and enable you to build workflows that mirror your entire business processes.

Let's say you want to build a flow that adds or updates customer details in CRM when a new order is created in your Shopify account. However, the **New order** trigger in Shopify does not provide the customer details. You can fetch the customer details using the **Fetch customer** action in Shopify, then use the data in other actions.



Another example of this scenario would be when a timesheet is created in TSheets, and you want to send an email to the related user. However, the **New timesheet entry** trigger provides only the user ID. To get other details, such as name and email, fetch the user by ID. Then you can use these details in the other actions of your flow.



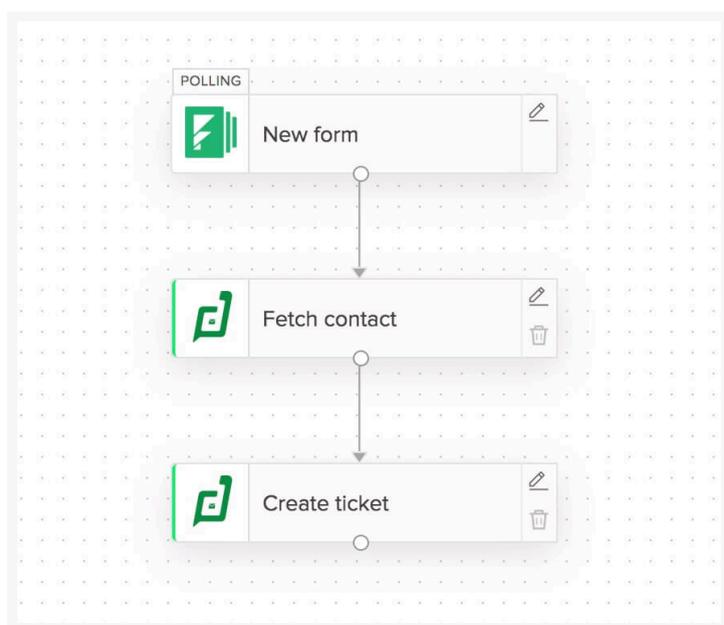
Note: When there are multiple records that relate to your search query, Zoho Flow will fetch the first record available.

Fetching related data

Sometimes you need to link records according to your process requirements. You can link records such as invoices with contacts, events with checklists, and leads with agents. Fetch actions get the related data that you need to link

records using a flow.

Let's look at an example. You build a flow to create a ticket in Zoho Desk based on a submission through a form. Zoho Desk allows you to create a ticket only if it is linked with a contact. So, when creating a ticket using Flow, *Contact ID* is a mandatory field. This is the ID of the contact in Zoho Desk, which will not be available from your trigger. You can fetch the contact by email or name before creating a ticket, then map the ID to the *Contact ID* field.



Note: Every app has its unique ID for an entity. For example, the ID of a particular contact in Zoho Desk will not be same as the ID of the same contact in any other app. If you're working with a record across different apps, remember to fetch the record details in that specific app.

Let's look at another flow that uses fetch action to link invoices with items. Consider that every time you receive a new order in your Shopify account, you want to create an invoice in your book-keeping app (Zoho Books, Quickbooks etc.). In this situation, you cannot select a particular item from the dropdown because the item will change for each order.

However, when you map the value to this field, you need the ID of this item in the accounting app (Zoho Books in this case). To get this ID you can use a fetch action before **Create sales invoice** action. With this ID, you can configure the flow to create an invoice for that specific item.

Create sales invoice
Creates a new sales invoice

Connection: Frank's Zoho Books account

Order number: \${trigger.order_number}

Item*: Use a Custom Value

Custom Value for Item*: \${fetchItemByName_1.item_id}

Item description: \${fetchItemByName_1.description}

Rate:

INSERT VARIABLE
Click fields to map to your action

- New order
- Fetch item - By name
 - Account name
 - HSN or SAC code
 - Image name
 - Is item taxable?
 - Is linked with Zoho CRM?
 - Item description
 - Item ID**
 - Item name

CANCEL DONE

Note: Most apps require the record ID (e.g.: Contact ID for contact) in the **Use a custom value** option in dropdowns. Some apps may accept name or email.

Creating or updating records

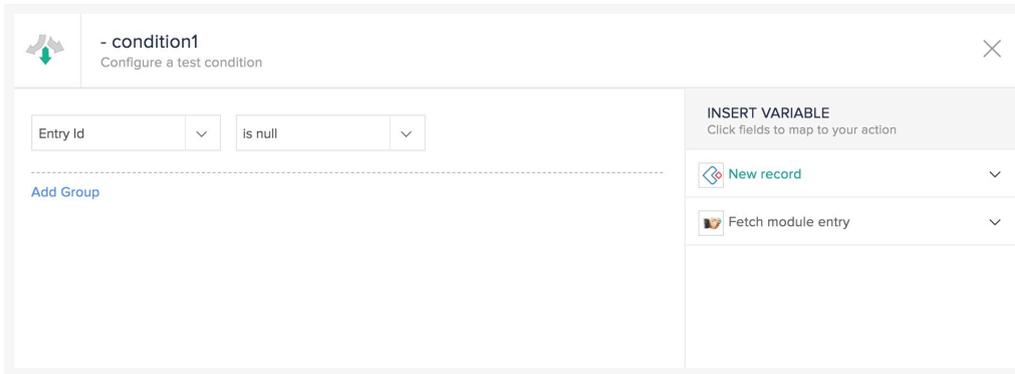
Fetch actions also help you check if a record exists in an app or not. Based on this, you can configure the flow to perform automated decisions. In this section, we'll talk about creating and updating records using fetch actions.

Create record

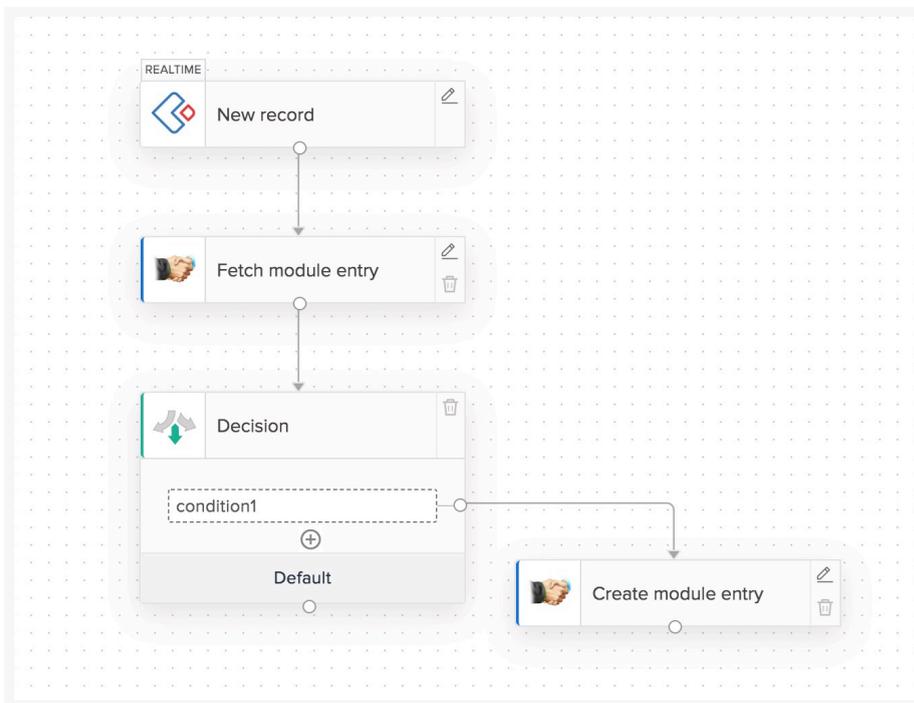
Suppose you have a web form where interested people can sign up for a free demo of your product. When someone signs up, you want to create a lead in CRM if the details don't exist.

This means that before creating a lead, the flow should check if the lead exists in your CRM. To do this, add the **Fetch module entry** action after the **New form submission** trigger.

Next, add a decision to check if the lead exists. In the decision, configure the **Entry ID is null** condition. Here, entry ID is a record ID.



Record ID is a unique identifier of a record. If the record exists, a record ID will also exist. If this condition is satisfied, it means that the ID doesn't exist, which means the lead doesn't exist. In this branch of the flow, add the **Create module entry** action and configure it to create a lead.



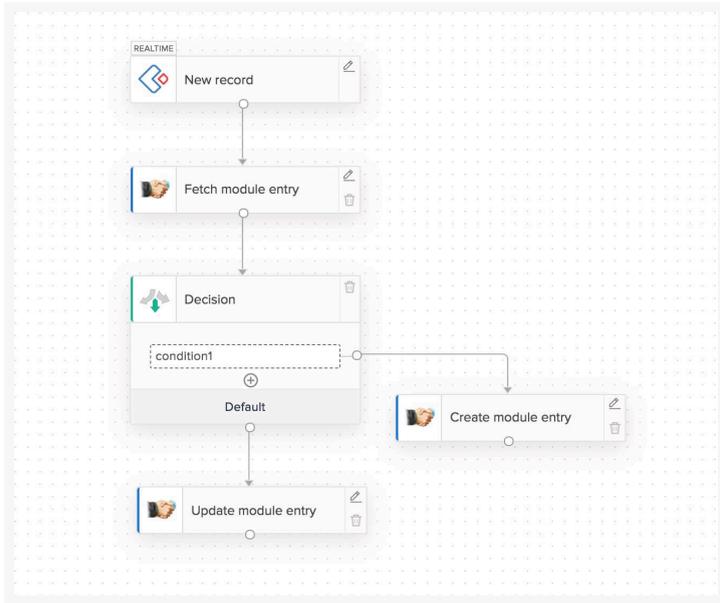
Some fetch actions have the option of creating a record if it doesn't exist already. When using such actions, a decision is not necessary.

E.g.: Fetch contact action in Zoho Books

Create or update record

Suppose you have multiple forms and want to track all the form submissions from a prospect. This means that the flow has to check if the lead exists, and decide if the details should be updated, or if a new lead must be created.

Again, let's use the fetch action and a decision with the **Entry ID is null** condition. If this condition is satisfied, add an action to create a record. If the condition fails, it means that the record exists. Add **Update module entry** action to the default decision branch.



Update record

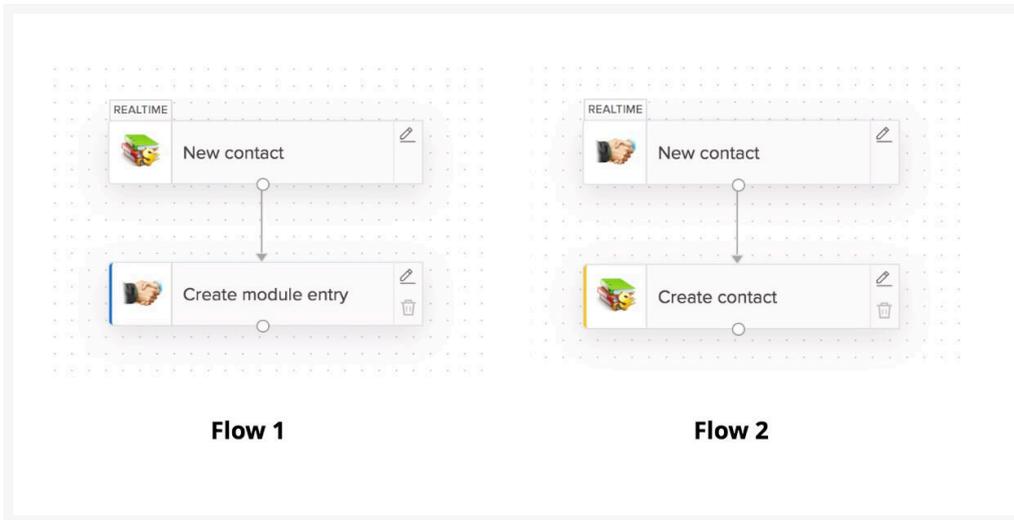
When updating (or deleting) a record, record ID is usually a mandatory field. This specifies the particular record that must be updated (or deleted). You can map this value from the fetch action.

Syncing data between apps

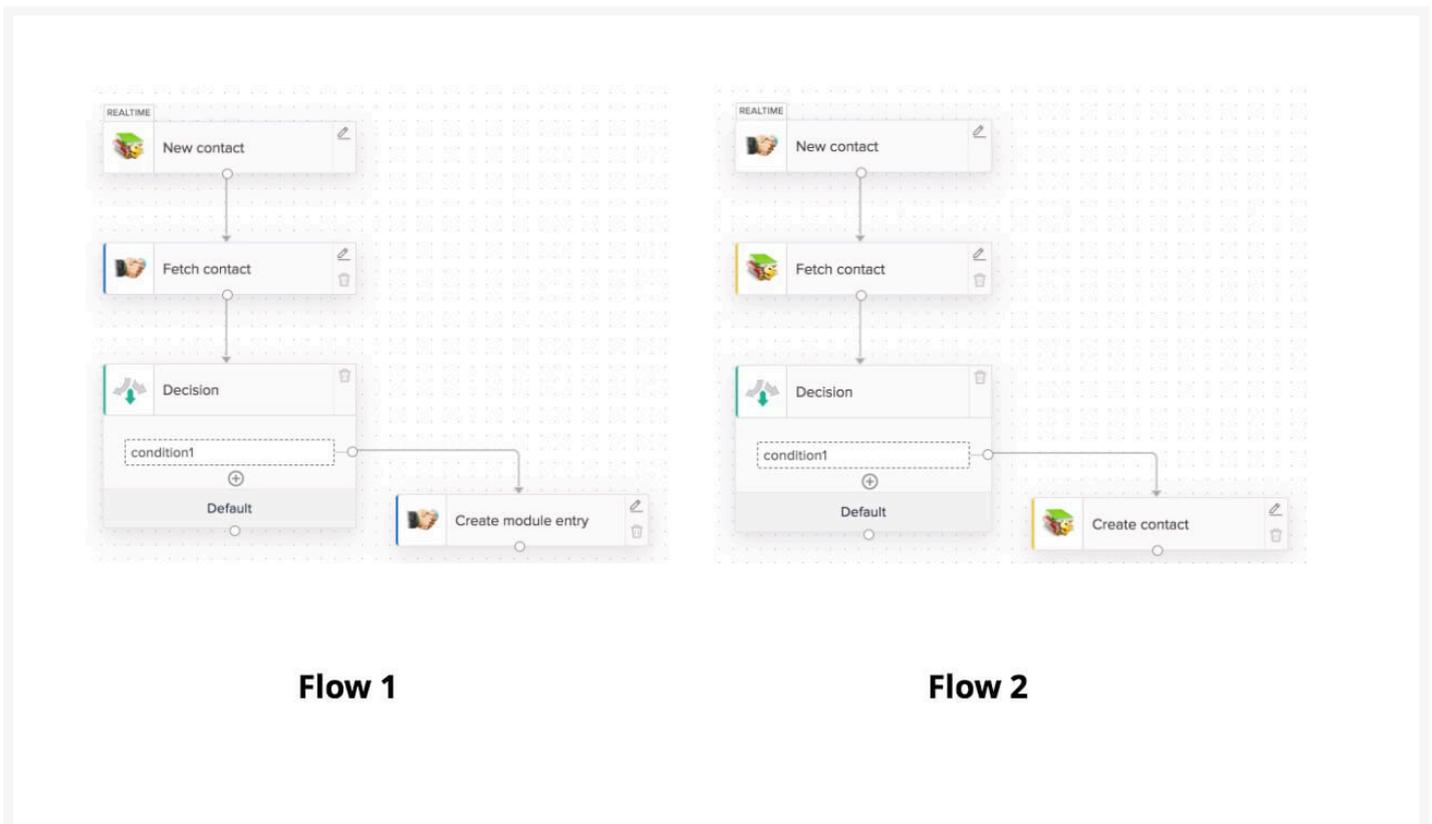
Keeping records in sync between apps is crucial for many business activities. Fetch actions can help you build workflows for this sync. For instance, your sales team works with Zoho CRM and your accounting team works with Zoho Books. The contacts in both these applications must be in sync for your business processes to run smoothly.

If you create two flows that are the reverse of each other, an infinite loop will be created.

When flow 1 is triggered, it will create a contact in CRM which will trigger flow 2. This will create a contact in Books and again trigger flow 1. This will go on creating duplicate contacts and exhausting your Flow tasks.



To prevent this from happening, add a fetch action and decision in each flow. Configure the flow to create contact only if it does not exist.



Now when flow 1 is triggered, it will create a contact in CRM and trigger flow 2. The fetch action in flow 2 will contain the record ID. The decision condition will fail and the flow will choose the default branch. So the contact won't be created again in Books.

This makes sure that the infinite loop is not created, saving you from duplicate contacts and wasted Flow tasks.

If you have questions or a use case to discuss, drop us an email at support@zohoflow.com