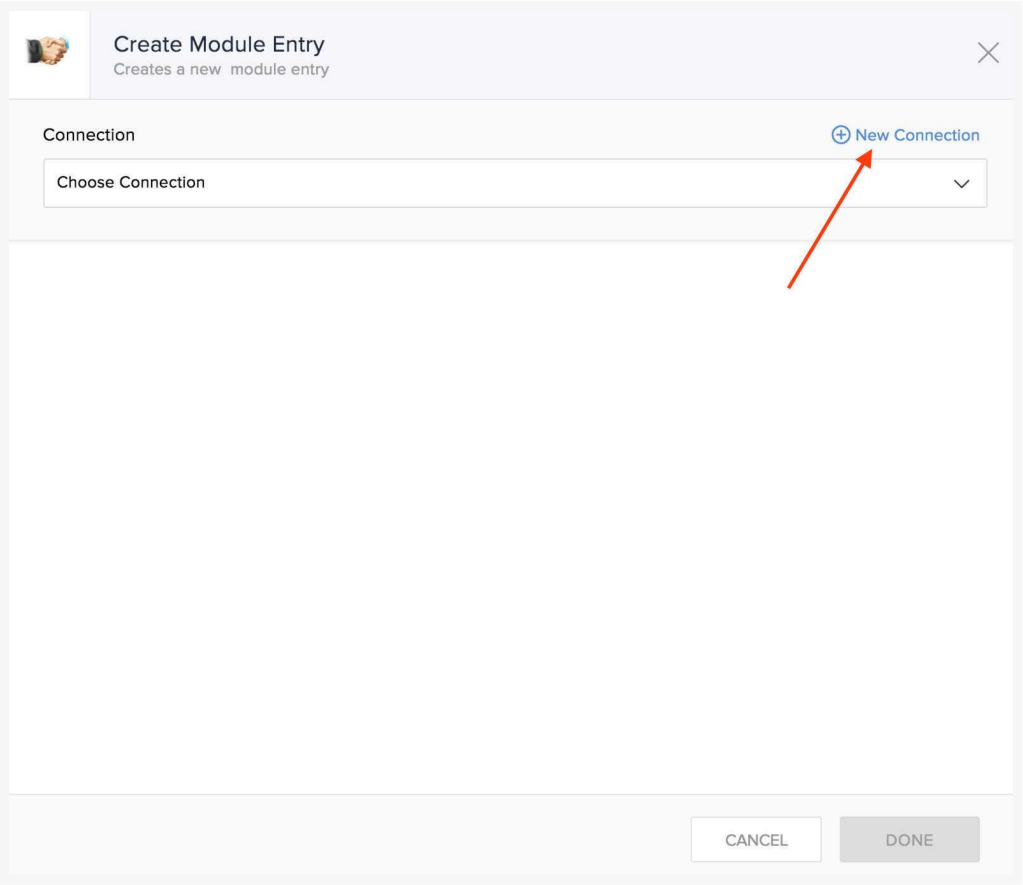
Zoho CRM

What is Zoho CRM?

Zoho CRM is on-demand customer relationship management software. You can customize your workflows and information, predict sales with detailed reports, and even monitor trends on social media and discover leads.

How to connect your Zoho CRM account to Zoho Flow

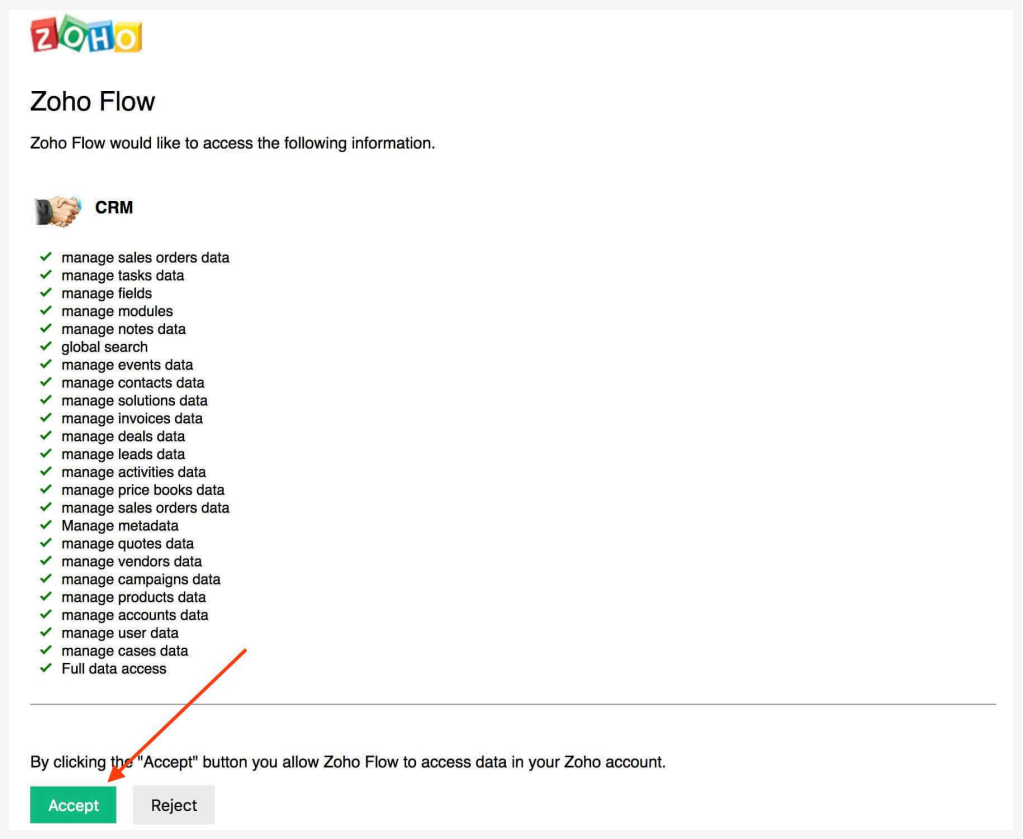
1. Select the required trigger or action. If you select a trigger, click **Next.**
2. If there are no other Zoho CRM connections in your account, click **Connect.** Otherwise, click **New connection.**



Alternatively, you can create a new connection by navigating to **Settings**, then **Connections.** Click **Create connection** and choose **Zoho CRM.**

3. Enter a **Connection name** and click **Authorize.**

4. Click **Accept** to allow Zoho Flow to access your Zoho CRM account.



API documentation

If you experience any Zoho CRM-related errors in your flows or if you wish to learn more about the Zoho CRM API, the API documentation can be found here.
Learn how to fix app-specific errors using API documentation

Triggers and actions available in Zoho Flow

General triggers and actions

Important note: The following triggers and actions mentioning "module" can be used for Zoho CRM's standard modules, such as Deals, Leads, Contacts, Products, Accounts, and Campaigns. They can also be used for custom modules that you have created in your account. When you set up these triggers or actions, you will be given an option to select the module you want to access.

Triggers

New module

This trigger lets you sync data with relevant applications, such as inventory, projects, or event management software.

New module entry

Use this trigger to send relevant documents when a deal is created, record purchase orders as expenses, and let your team know when new tasks are added.

Updated module entry

This trigger lets you update the details of a contact in your helpdesk software, keep your inventory updated with product details, and message your marketing team when a campaign is updated.

Actions

Create module entry

This action lets you create sales orders, purchase orders, and invoices when they are approved in your accounting application.

Create or update module entry

Sync estimates with your accounting app, create events from your scheduling tool, and update product details from inventory records with this action.

Fetch module entry

Use this action to fetch the details of a module entry before updating it.

Update module entry

Update the details of sales orders, events, and cases using this action.

Update related module entry

This action helps you link a lead to an activity, a call to a note, or an account to a case.

Delete module entry

Use this action to delete customer details based on entries through your request forms.

Specific to Contacts

Triggers

New contact
Use this trigger to assign an agent to the contact, remind the agent to contact the customer periodically, and add the contact to your book-keeping system.

Actions

Create or update contact
Build customer profiles and keep your CRM data up to date with this action.

Note: This action creates or updates a contact based on email. If the email exists, the contact details are updated. Otherwise, a new contact is created.

Fetch contact
Fetch details of the contact to associate it with deals.

Convert lead to contact
Use this action to convert leads when they place an order or schedule a meeting with your team.

Specific to Leads

Triggers

New lead
Add new leads to your contact manager tool or spreadsheet, and reach out to them via email using this trigger.

Actions

Create or update lead
Create or update lead details based on social media interactions, emails, and event registrations.

Note: This action creates or updates a lead based on email. If the email exists, the lead details are updated. Otherwise, a new lead is created.

Fetch lead
Fetch details of the lead using this action to send personalized messages.

Convert lead to contact

Use this action to convert leads when they place an order or schedule a meeting with your team.

Specific to Deals

Triggers

Updated deal

When a deal is updated, send a confirmation email to the customer, notify different teams through chat, and update the details in your project management tool using this trigger.

Actions

Fetch deal

Use this action to fetch all the details about a deal.

Other triggers and actions

Triggers

New note

Use this trigger to notify yourself and your team when a note is added.

New user

When a new user is added in your CRM account, add them to your campaign, accounting, or helpdesk software with the help of an automated decision.

Actions

Add note

Add notes from emails and surveys to the specific record with this action.

Fetch account

When a deal or order is created, fetch details of the account to check for loyalty points and provide discounts.

Fetch event

This action lets you check if the event exists in your CRM system when an event is created in your event management software.

Fetch product

Fetch details of the product to check if the product exists and configure your flow to make automated decisions.

Fetch user

When a user is created in your helpdesk software, use this action to check if the user details have been added in your CRM.

Send email template

Use this action to send templated emails to leads, customers, or contacts.

Send mail merge

Send a mail merge with details of a record using this action.

Remove tags

This action helps you remove the tags on a selected record.

Raise signal

Use this action to raise a real-time notification from your leads, contacts, or potential customers to your Zoho CRM account.

Creating and updating module entries

Create module entry

To create a record in the events module with this action, use the following custom function. This function modifies the date fields to the format required by the events module of Zoho CRM.

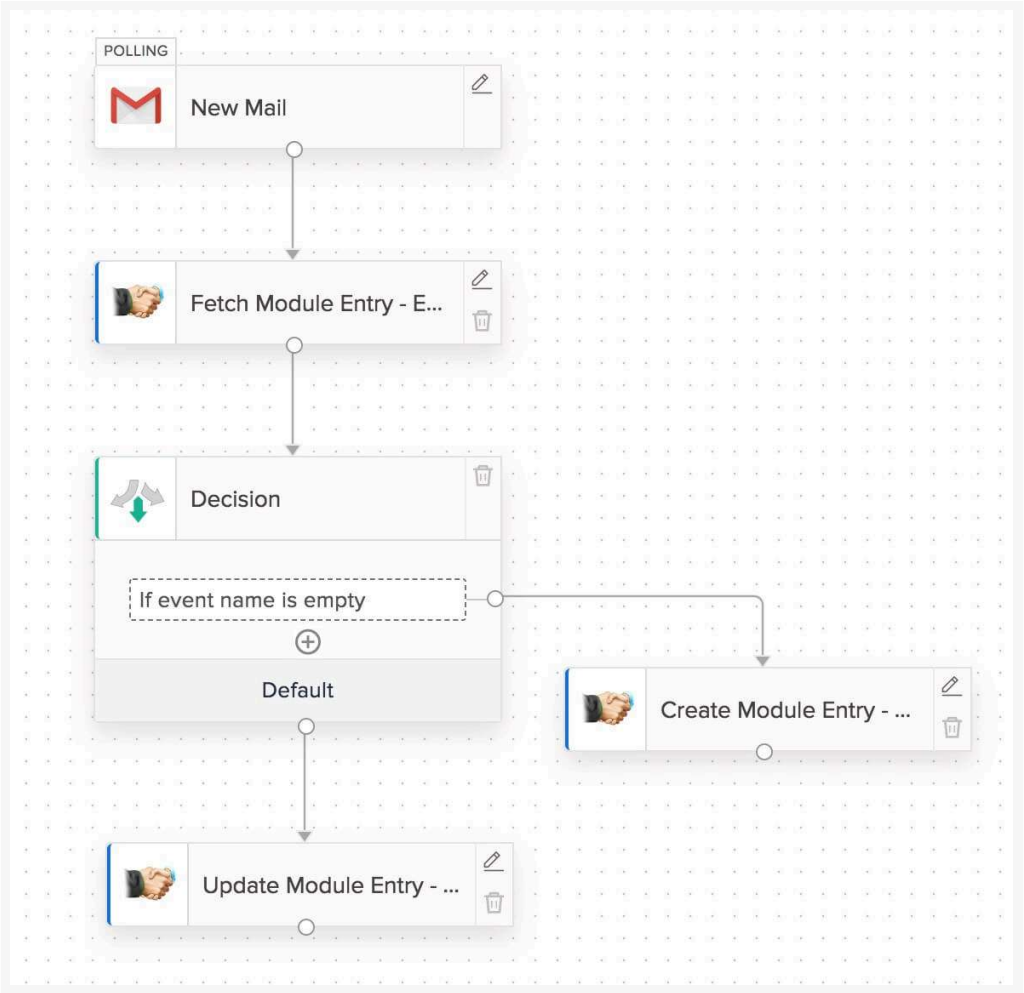
```
string dateFormatISO(string myDate, string fromFormat)
{
mydateObj = myDate.toTime(fromFormat);
resultDate = mydateObj.toString("yyyy-MM-dd'T'HH:mm:ss'Z'");
return resultDate;
}
```

Create or update module entry

This action creates a new entry in the selected module. If the entry already exists, the action updates the entry details. Zoho CRM checks for duplicate records using the following criteria.

Module name	Criteria
Leads	Email
Accounts	Account name
Contacts	Email
Deals	Deal name
Campaigns	Campaign name
Cases	Subject
Solutions	Solution title
Products	Product name
Vendors	Vendor name
Price books	Price book name
Quotes	Subject
Sales orders	Subject
Purchase orders	Subject
Invoices	Subject
Custom modules	Name

Note: Zoho CRM's API does not support create or update as a single action for tasks, calls, or events. Use the **Fetch module entry** action to fetch variables from the module to map. Then configure a decision to check if a particular field is empty. If it is empty, this means the record does not exist. You can then branch your flow using **Create module entry** and **Update module entry**.



Working with unique fields in a module

In Zoho CRM modules, a field becomes a unique field when you have enabled duplicate check on that field. For such fields, if a duplicate value is passed using an action such as *Create module entry*, the flow would fail. In your Zoho CRM **Module Settings**, disable **duplicate checking** for the required field if you think it need not be a unique field.

Otherwise, you can use a *Fetch* action and *Decision* logic to:

1. Filter out duplicate entries and avoid passing them on to a *Create* action
2. Check if the record already exists, and use an *Update* action to update the existing record with the new information.

Triggering native workflow rules in Zoho CRM

In generic module based actions, such as *Create module entry*, the **Trigger for** option lets you choose whether the action should trigger *workflow, approval,* or *blueprint* rules in Zoho CRM. Choose **None** if you don't want the action to trigger any rules. If you leave the option unattended, the default option, **All of the above** will be in effect.

However, this option is not available for the specific module based actions, such as *Create or update lead*, and will always trigger any existing workflow rules you have set up in Zoho CRM.

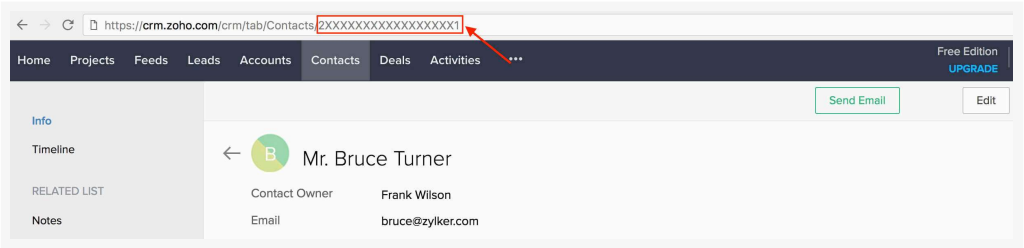
Working with the Add notes action

While configuring the **Add notes** action on Zoho Flow, you need to enter a value for **Entity ID**. This is a unique ID associated with the entity (record) in Zoho CRM. To get this ID, go to your Zoho CRM account and select the module and entity you want to add the note to. Your URL will be in the format:

https://crm.zoho.com/crm/tab/{module_name}/**{Entity_ID}**.

Use the value in the place of **{Entity_ID}** to fill in the entity ID field in your configuration.

For example, if you select the Contacts module and click on a particular contact, your URL will look something like: https://crm.zoho.com/crm/tab/Contacts/2XXXXXXXXXXXXXXXXX1. Your entity ID is: 2XXXXXXXXXXXXXXXXX1



Common problems

1. **I can't enter boolean values as inputs.**
   Use **${false}** and **${true}** to add boolean values as the input in your configuration.

2. **My flow with Zoho CRM as the trigger app does not start.**
   Zoho CRM triggers will not start your flow if the records are imported or if data updates occur

through Zoho CRM's custom functions.

3. **I get the error message "Module limit exceeded."**
   Limits in Zoho CRM depend on your subscription level. Check this link for more details. You can delete some of your existing workflow rules (Zoho CRM -> Settings (gear icon) -> Automation -> Workflow Rules) and try to switch on the flow again. Alternatively, you can write to Zoho CRM support (support@zohocrm.com) to increase your workflow limit in CRM.

4. **What value should I enter in the 'Owner' field when working with module entries?**
   This field only accepts an ID value. You can use the **Fetch user** action to get the user ID from CRM and then map it to the 'Owner' field.

5. **I get the error message "The expected input type is jsonarray"**
   Sometimes, a variable holding an array can get auto-converted to a string datatype. In such cases, when you insert that variable into a field that requires a JSON array, you will encounter this error. In such cases, you can pass the string or Multi-select pick list values as a JSON array using the following custom function:

```
list convertToList(string listStr)
{
return listStr.toList(",");
}
```

   The above custom function will create a JSON array by looking for values that are separated using comma (eg., a,b,c). If you want to do the same from a semicolon-separated string (eg., a; b; c), replace the comma in the third line (listStr.toList(",");) with a semicolon.