



Working with date and time values in Zoho Flow

Different apps talk about date and time values in different ways. American, European, International—there are so many date formats, that it's a challenge to transfer date-time values across apps automatically. Some apps may not accept other date formats, while others may misinterpret the value. For example, if an app receives an input value as 04-03-2020, it may understand it as 4 March 2020, or 3 April 2020, depending on its configuration.

When you build flows, you need to make sure that date-time values are converted accurately. This minimizes errors in your flow and ensures that the right values are conveyed to other apps. In this article, we'll learn how to identify, convert, and work with various date formats that your apps provide. Before we dive in to converting date formats in flows, let's look at the basic notations and formats we'll be using.

Before we dive in to converting date formats in flows, let's look at the basic notations and formats we'll be using.

Notations and formats

Basic notations

Letter	Date or Time component	Examples
y	Year	1996; 96
Y	Week year	2009; 09
M	Month in year	April, Apr, 04
w	Week in year	18
W	Week in month	3
d	Day in month	10
D	Day in year	211
E	Day name in week	Monday; Mon
a	AM or PM marker	PM
H	Hour in day (0-23)	2
h	Hour in AM/PM (1-12)	3
k	Hour in day (1-24)	21
K	Hour in AM/PM (0-11)	10
m	Minute in hour	54

s	Second in minute	30
S	Millisecond	853
X	ISO 8601 timezone	-08; -0800; -08:00

Common date and time formats

ISO 8601

This is a standardized way to represent date and time values, and is the most common format in which date-time is communicated across apps.

Date and time format: **yyyy-MM-ddThh:mm:ssXXX**

E.g.: 2018-11-26T11:54:39+05:30

Date format: **yyyy-MM-dd**

E.g.: 2018-11-26

Unix timestamp

This represents the number of seconds elapsed since 1 January 1970 (excluding leap seconds)

E.g.: 1546491981

American

Format: **MM/dd/yyyy**

E.g.: 11/26/2018

European

Format: **dd/MM/yy**

E.g.: 26/11/2018

International

Format: **yyyy-MM-dd**

E.g.: 2018-11-26

Important points to note

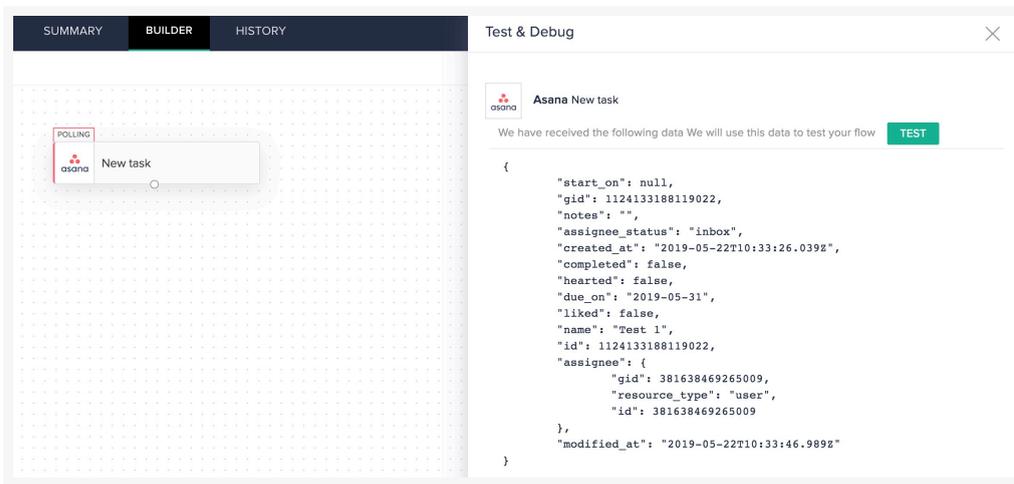
- UTC, GMT, etc. are ways of tracking time. For all practical scenarios, UTC and GMT will be the same.
- ISO 8601, RFC 1123, etc. are date-time formats. ISO 8601 is the accepted format in all the modern applications.
- The character **T** in the middle separates the year-month-day part from the hour-minute-second part. It should be parsed from the date-time value that you want in your applications. Use single quotes to parse it from the input value. For example, "**yyyy-MM-dd'T'hh:mm:ss**" should be given as the output format and not "**yyyy-MM-ddThh:mm:ss**".

- Use single quotes(' ') to parse any set of characters from a string that is given inside double quotes (" ").
- The character 'Z' at the end of a date-time format means the date-time is UTC. That is, an offset-from-UTC of zero hours-minutes-seconds.
- Instead of 'Z', if you see '+01:00', it means the timezone is one hour ahead of UTC at the moment, as noted by '+01:00'

Identify date formats in flows

When apps send data through APIs, they always use a specific format. You can look at the app's API documentation to identify the date format it uses. Some apps use different formats for different entities. Make sure you refer to the date format in the right section of the documentation.

Another way to find the date format is by looking at the data coming in to your flow. When you test the trigger, the output will contain the date value.

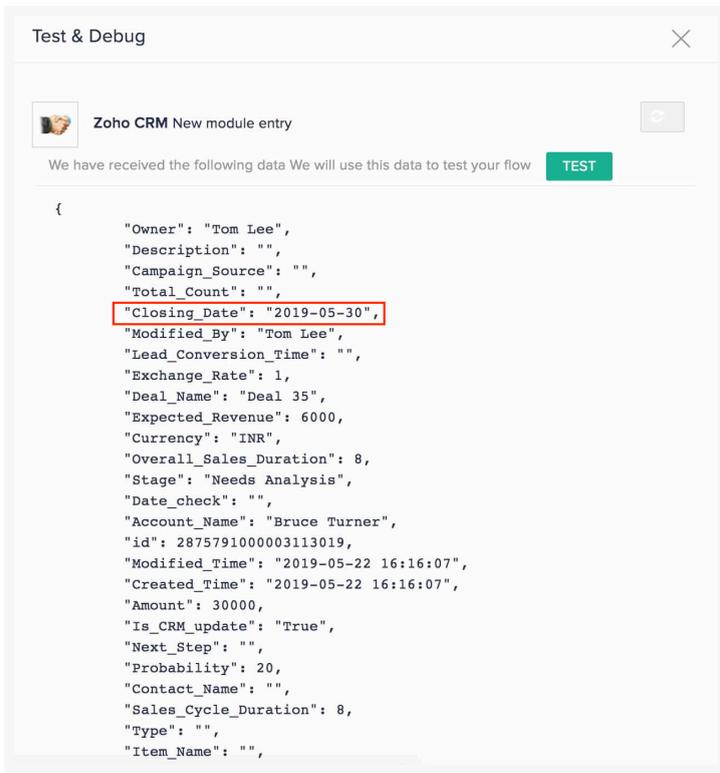


The screenshot shows the Zoho Flow Builder interface. On the left, there is a canvas with a 'POLLING' trigger connected to an 'Asana New task' action. On the right, a 'Test & Debug' window is open, displaying the data received from the trigger. The data is a JSON object with the following structure:

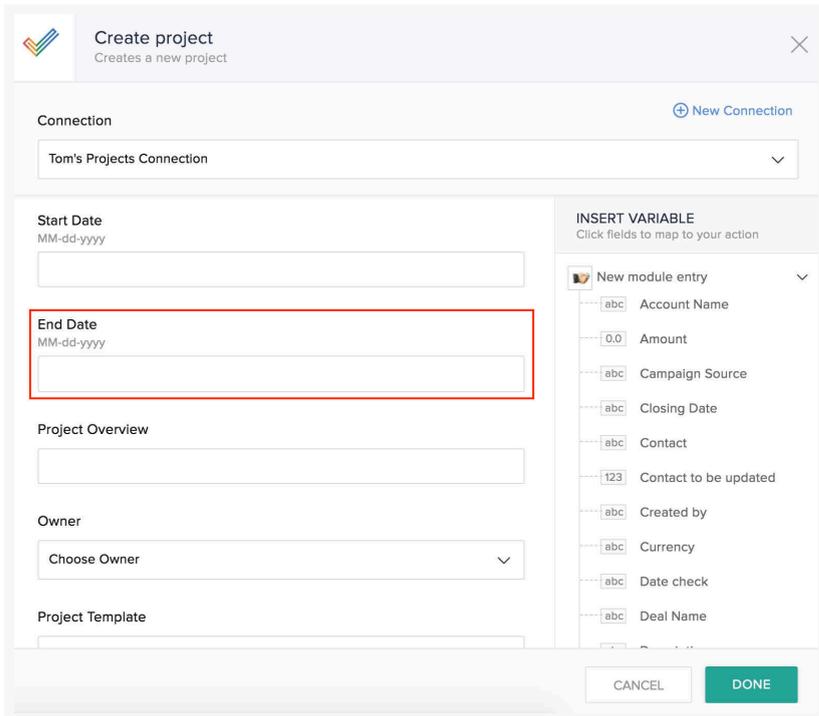
```
{
  "start_on": null,
  "gid": 1124133188119022,
  "notes": "",
  "assignee_status": "inbox",
  "created_at": "2019-05-22T10:33:26.039Z",
  "completed": false,
  "hearted": false,
  "due_on": "2019-05-31",
  "liked": false,
  "name": "Test 1",
  "id": 1124133188119022,
  "assignee": {
    "gid": 381638469265009,
    "resource_type": "user",
    "id": 381638469265009
  },
  "modified_at": "2019-05-22T10:33:46.989Z"
}
```

When you configure actions, we try to display the accepted date format as much as possible.

Consider a flow that creates a project in Zoho Projects when a deal is created in Zoho CRM. When we look at the data CRM sends for a new deal, we see that the date is in yyyy-MM-dd format.



While configuring the action in Zoho Projects, we see that it accepts only MM-dd-yyyy format.



Because of this inconsistency, we can't map the date values directly from CRM to Projects. Now that we've identified what we need to work with, let's look at how to convert the date value.

Convert date formats

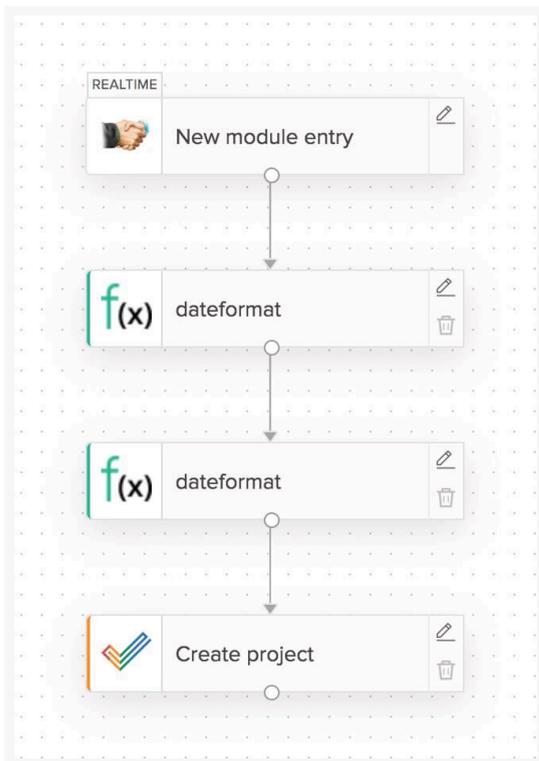
To convert date from CRM to Project's format, we use a [custom function](#) between the trigger and action.

Use this code to create your custom function:

```
1. string dateformat(string dateStr, string inputFormat, string outputFormat)
2. {
3.   myDate = dateStr.toTime(inputFormat);
4.   return myDate.toString(outputFormat);
5. }
```

This function requires three inputs: the date from CRM trigger, its format, and the format accepted by Projects. This custom function can convert only one date and time value during an execution.

In our flow, we'll use the custom function between the trigger and the action. We need to map two date values: start and end date. So we use the custom function twice and map the output of the functions to the respective fields in Zoho Projects.



You can also include the date format in the custom function code, if you're going to use the function for a specific case.

```
1. string dateFormatCal(string dateStr)
2. {
```

```
3. myDate = dateStr.getTime("yyyy-MM-dd HH:mm:ss");
4. return myDate.toString("MM-dd-yyyy");
5. }
```

This function converts any date format to the standard format of **yyyy-MM-dd HH:mm:ss**, then to the specified format (MM-dd-yyyy in this case). If you choose to use this kind of function, you need to write a different function for each output date format.

Let's take another example of date-time format conversion between Zoho CRM and Zoho Desk. Suppose the date-time value coming from Zoho CRM's output is **2021-07-20T10:33:00+05:30**, and you want it to be converted to Zoho Desk's format, which should look like **2021-07-20T10:33:00.000Z**.

To do this, you should configure your custom function in the following manner:

```
1. string dateFormat(string dateStr, string fromFormat, string toFormat)
2. {
3.     dateObj = dateStr.getTime(" yyyy-MM-dd'T'HH:mm:ss");
4.     return dateObj.toString("yyyy-MM-dd'T'HH:mm:ss.SSS'Z'");
5. }
```

Add or subtract date-time values

When you work across countries, you need to make sure your flows work with varying time zones. Custom functions help convey these date-time values across time zones accurately.

You can also use custom functions to add or subtract date-time values and set reminders. To understand how to do this, let's build a flow to notify your team on Zoho Cliq a day before a meeting scheduled in your Google Calendar.

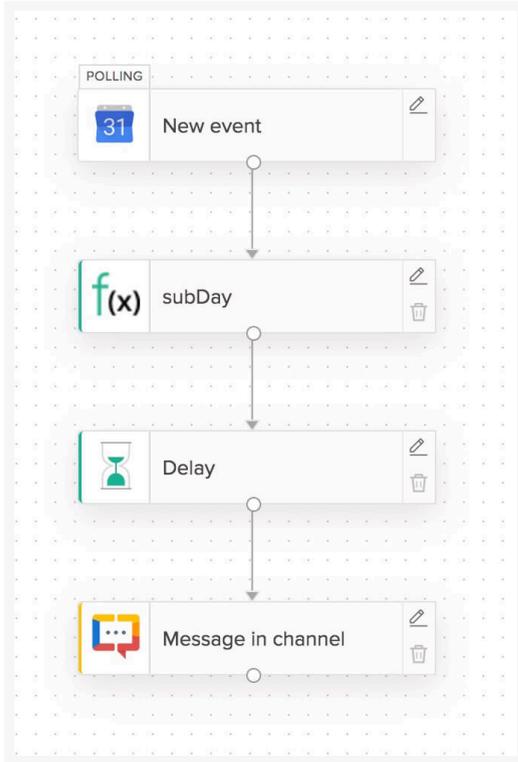
We'll use this custom function to calculate the day before the event:

```
1. string subDay(string dateStr)
2. {
3.     dateObj = dateStr.toDate("yyyy-MM-dd'T'HH:mm:ss");
4.     resultDate = dateObj.subDay(1);
5.     return resultDate.toString("yyyy-MM-dd");
```

6. }

The date of the meeting is the input of this custom function. This function then calculates the date before the day of the meeting.

Now we add a delay that stalls the flow until the date provided by the custom function. The flow then sends a reminder to the team.



Depending on your requirements, you can modify these functions or write your own. Refer to the [Deluge help page](#) to learn more about functions you can use.

If you'd like to talk with us about date and time formats, shoot us an email at support@zohoflow.com.