

## Posting updates to Microsoft Teams

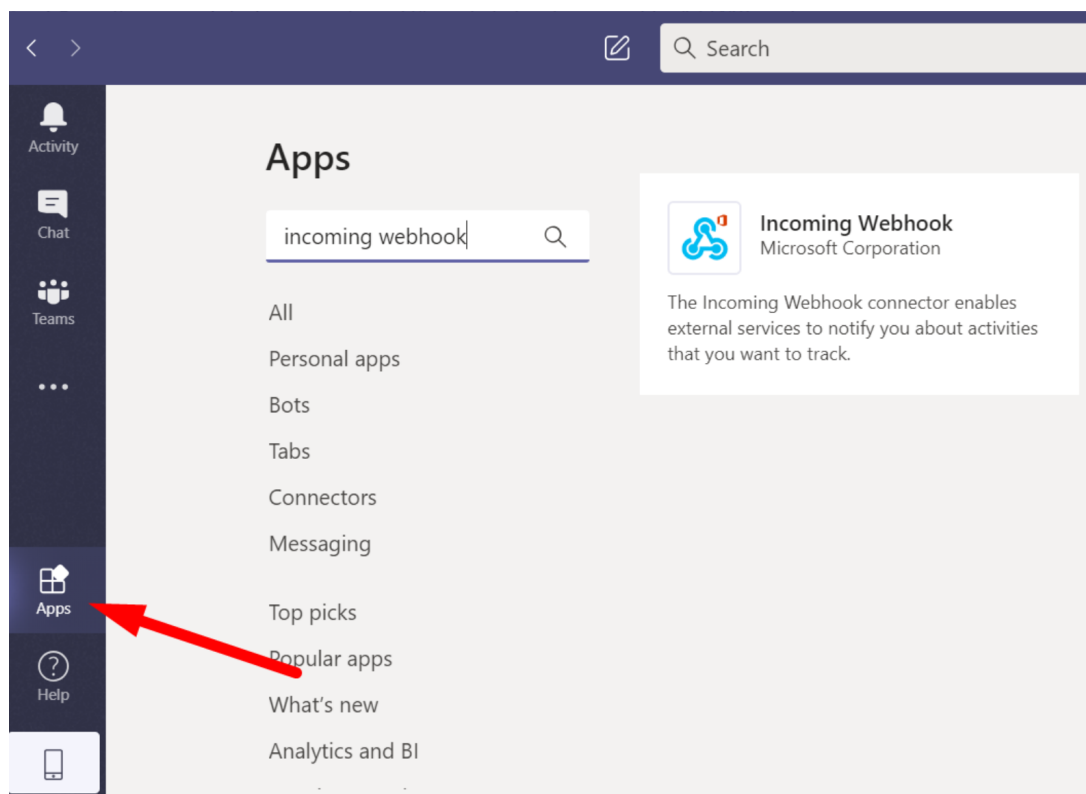
Matthew Wray - 2025-02-10 - Comments (0) - Using Deskpro

You can post ticket updates directly to Microsoft Teams from Deskpro.

To do this you simply need to setup an incoming webhook in Teams and a trigger in Deskpro.

### Creating an Incoming Webhook in Teams

To create an incoming webhook in Teams navigate to apps in the bottom left of the UI then search for Incoming Webhook.




Select Incoming webhook -> Add to team and then select your team and the channel you wish to post to and hit Set up connector.

On the setup page you will need to create a name for your webhook and you also have the

option to select an image that will appear next to posts made by the webhook into teams.

Hit create and the webhook will be generated:

Connectors for "General" channel in "Deskpro" team ×

 Incoming Webhook Send feedback


The Incoming Webhook connector enables external services to notify you about activities that you want to track. To use this connector, you'll need to create certain settings on the other service, which needs to support a webhook that's compatible with the [Office 365 connector format](#).

Fields marked with \* are mandatory


Enter a name for your IncomingWebhook connection. \*

Customize the image to associate with the data from this Incoming Webhook.

[Upload Image](#)



Copy the URL below to save it to the clipboard, then select Save. You'll need this URL when you go to the service that you want to send data to your group.

[Done](#) [Remove](#)

Note: If you're a software developer and want to learn more about sending data to Office 365 using Incoming Webhook, see [Get started with Office 365 Connector Cards](#).

Copy the URL created and hit done.

## Creating the trigger in Deskpro

In this example we're going to create a trigger that posts a message to our Teams channel

when a new ticket is created.

Go to Deskpro and then Admin > Tickets > New ticket Triggers and hit Add.

With this example, we want to post with all tickets so we'll leave all events checked and won't specify any criteria.

For our action we'll select 'Call Web Hook':

The screenshot shows the 'Call webhook' configuration form. The form is titled 'Call webhook' and is part of a sequence of actions. The 'Then' section indicates that the following actions will run. The form fields are as follows:

- Webhook URL: `https://www.office.com/webhook.office.com/webhookb2/...` (partially redacted)
- Username: (empty)
- Password: (empty)
- Method: `POST`
- Custom headers: (empty)
- Custom data: `{}`
- Payload type: `json`
- Timeout: `30` Seconds
- Certificate validation:
- Only custom payload:

We then need to configure our action.

We'll need to define the following fields:

- Webhook URL: Set this to the webhook you created in Teams
- Method: Set this to POST
- Custom data: We'll come to this in a moment
- Payload type: json
- Timeout: I've set this to 30 seconds
- Certificate Validation: Ensure this is disabled
- Only custom payload

## Defining the Custom data field

You can read in detail about the format data needs to be sent in the [Microsoft documentation](#)

In my example though I'm just going to create a simple post with the user name, ticket id, ticket subject and a link to the ticket.

The code required for this is as below **(be sure to update the URL in the potential action target from [example.deskpro.com](#) to your helpdesk URL)**:

```
{

  "@type": "MessageCard",
  "@context": "http://schema.org/extensions",
  "summary": "Deskpro ticket",
  "themeColor": "0078D7",
  "title": "New Deskpro ticket",
  "sections": [

    {

      "activityTitle": "User name",
      "activitySubtitle": "{{ ticket.person.display_name }}",
      "facts": [

        {

          "name": "Ticket ID",
          "value": " {{ ticket.id }}"

        },

        {

          "name": "Subject",
          "value": " {{ticket.subject }}"

        }

      ],

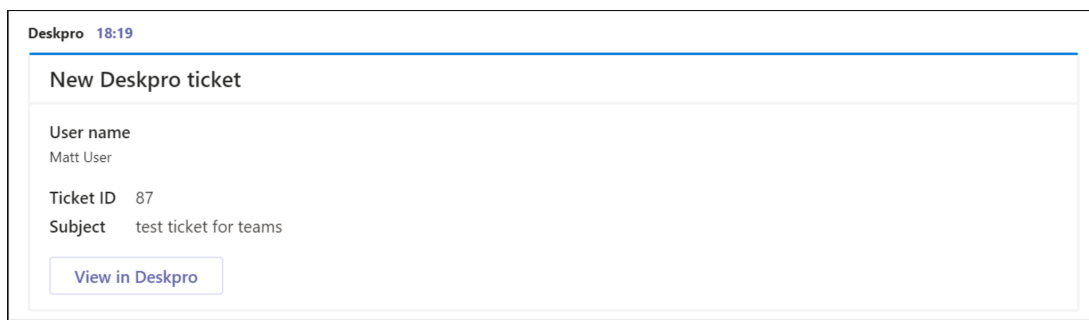
    },

  ],

}
```

```
    "message": "Message content"
  }
],
"potentialAction": [
  {
    "@type": "OpenUri",
    "name": "View in Deskpro",
    "targets": [
      { "os": "default", "uri":
"https://example.deskpro.com/app#/t/ticket/{{ ticket.id }}" }
    ]
  }
]
}
```

Here you can see we've used variables for username, ticket ID, and subject and will post the below in teams (including a button that links to the ticket):



You can read about other potential variables you could use in the [Deskpro guides](#).

And you can refer to the [Microsoft documentation](#) on how to structure your JSON payload in different ways.