How can I automatically change the ticket status when a comment is added in JIRA (v3)?

Ashley Dawson - 2022-07-04 - Commenti (0) - Deskpro Apps

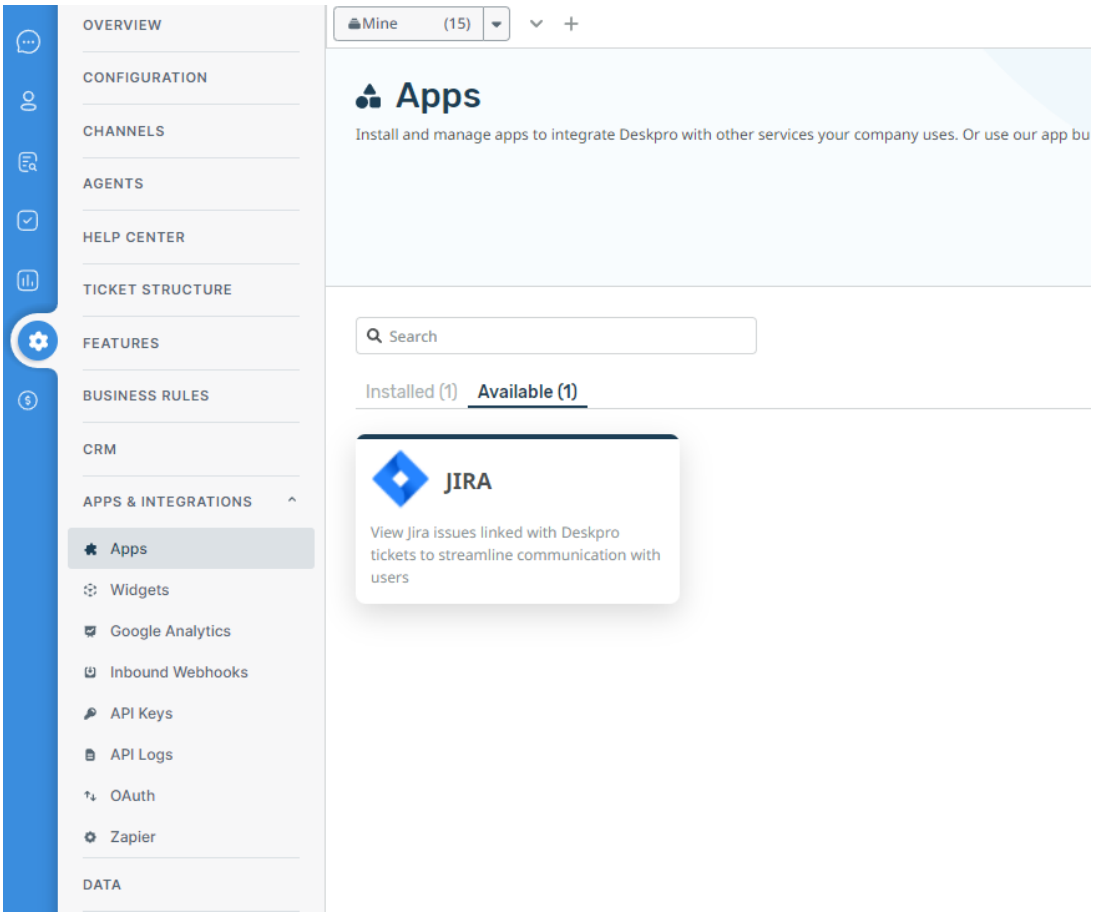
In this article, we're going to walk through the process of setting up a Webhook to receive events from JIRA that will "tell" Deskpro to affect a ticket in some way. For example, we're going to listen for when a JIRA comment is added and update the Deskpro ticket status.

Why is this useful?

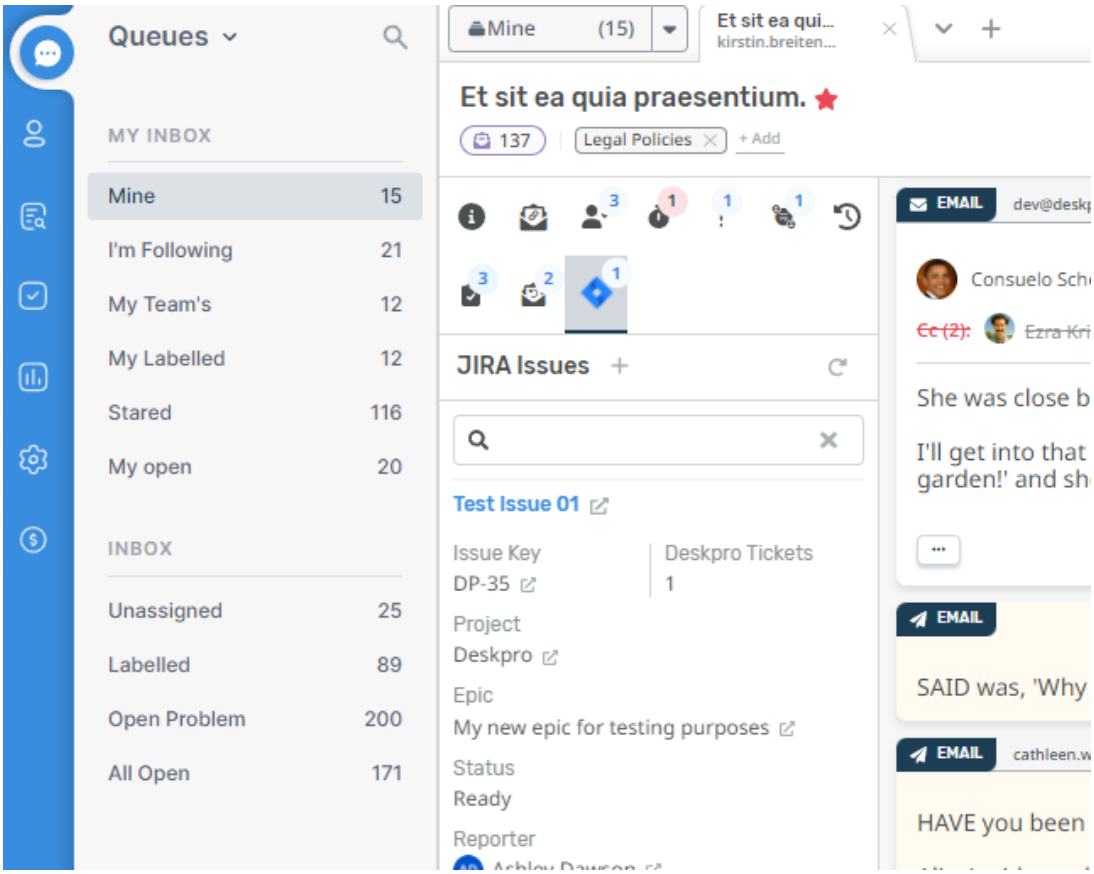
Listening to events from other systems is very useful for keeping Deskpro up-to-date with recent changes that were not actioned through Deskpro itself. You can imagine listening to when an issue is updated in JIRA and then adding a note or changing the status of an associated Deskpro ticket.

Installing the JIRA app

The first step is to install the JIRA app. To do this, navigate to the "Admin > Apps & Integrations > Apps" section of Deskpro.

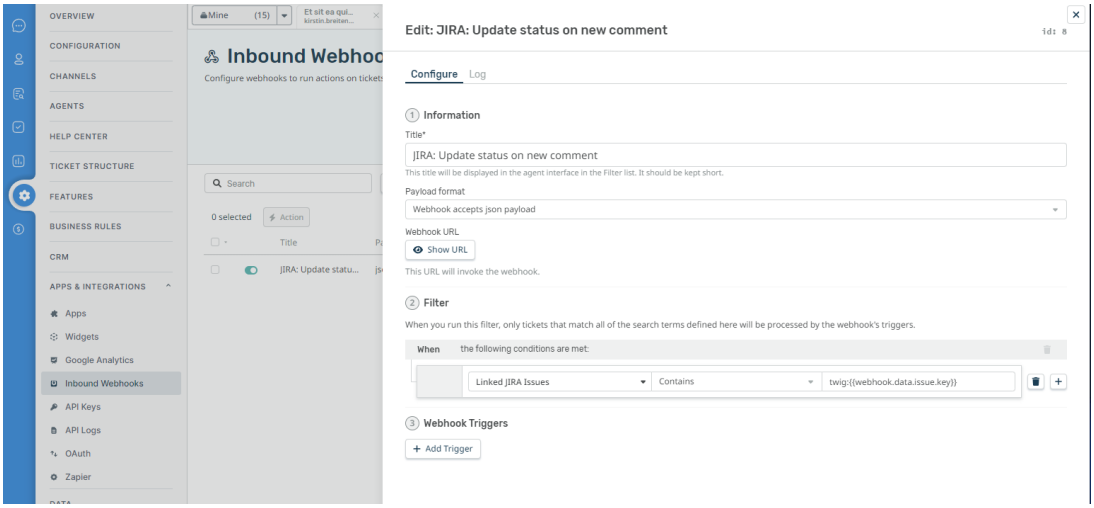


Click on the JIRA app and install it using your JIRA API credentials. Once the app is installed, go to one of your Deskpro tickets and use the JIRA app to link a JIRA issue to your Deskpro ticket.



Creating the Webhook

Now that we have a linked issue, let's add a webhook to "listen" to JIRA events. Navigate to the "Admin > Apps & Integrations > Inbound Webhooks" section and add a new webhook.



Add the following settings to your new webhook:

- **Title**: "JIRA: Update status on new comment" - this is the title of your webhook, it's important to give it a descriptive name so you can tell what the webhook is doing at

a glance

- **Payload Format**: “Webhook accepts json payload“ - JIRA will send data in JSON format, let’s set our webhook up to parse this type of data
- **Filter**: “Linked JIRA Issues -- Contains -- twig:{{webhook.data.issue.key}}“ - our filter is going to interrogate the data sent from JIRA and attempt to match the JIRA issue key with the linked Deskpro tickets

Click “Create“ to create the new webhook before we continue

Webhook Trigger and Action

Now that we can listen to and match Deskpro tickets from JIRA events, the next step is to actually “do“ something in Deskpro. In this case, we’re going to update the ticket status to “Awaiting Agent“ if a new comment is added to the associated issue in JIRA itself.

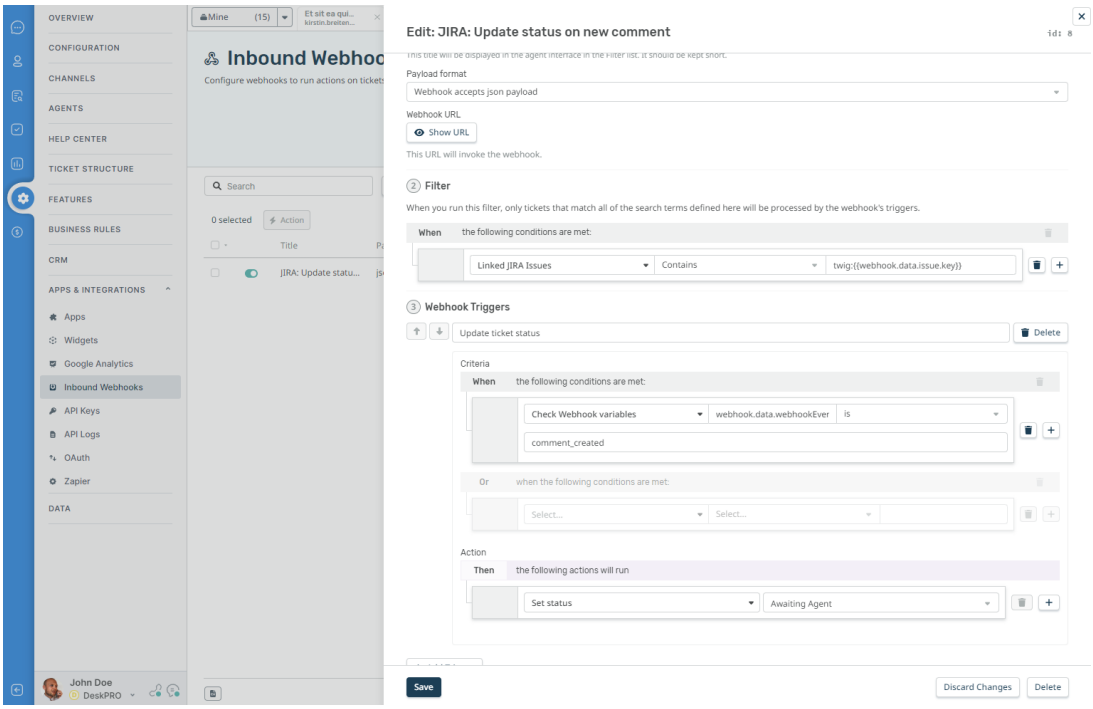
Reopen the webhook and add a webhook “trigger“ and set the criteria to:

- **Operation**: Check Webhook variables
- **Field**: webhook.data.webhookEvent
- **Operator**: is
- **Value**: comment_created

Also, let’s add the “action“ that will be performed:

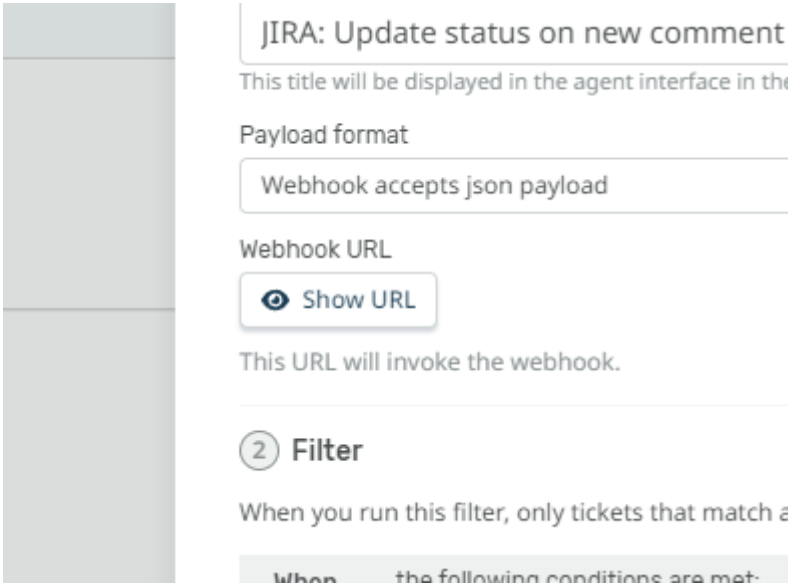
- **Type**: Set status
- **Value**: Awaiting Agent

Now click “Save“



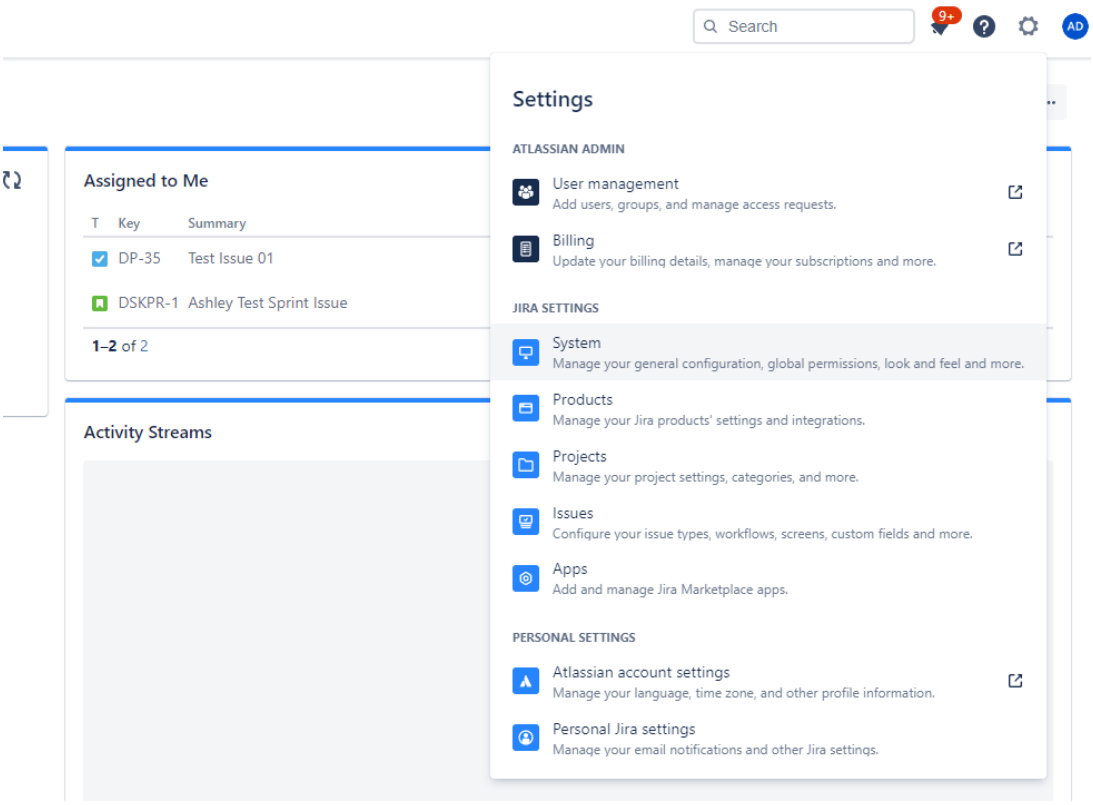
Ok great! We now have our webhook in Deskpro. Next, let's let JIRA know about our new webhook URL so it can start sending us events.

Grab the webhook URL by revealing it by clicking "Show URL" and copying it to your clipboard.

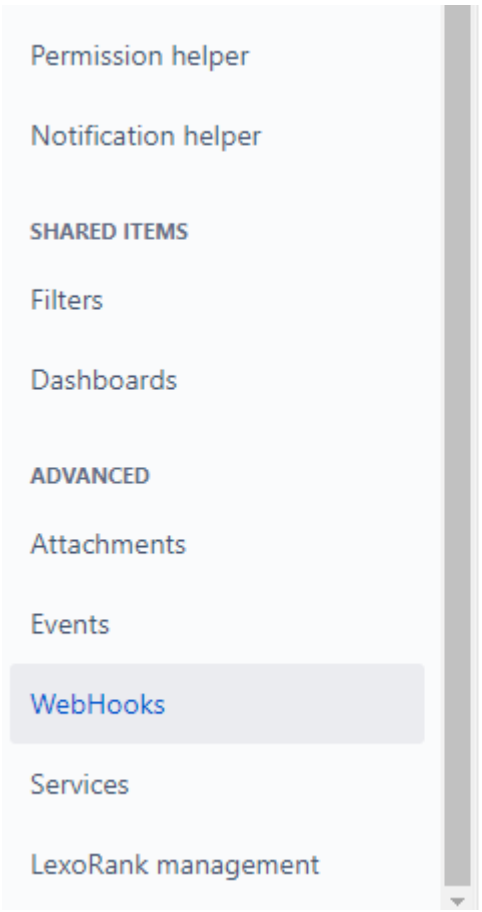


Adding the webhook to JIRA

Login to your JIRA account and navigate to the "JIRA Settings > System" section in the cog menu at the top right of the screen.



Next, navigate to the “WebHooks“ section in the left-hand sidebar.



Create a new webhook and enter the following details:

- **Name**: “Deskpro Status Change Webhook“ - a descriptive name so we can recognize our webhook at a glance
- **Status**: “Enabled“ - enable this webhook so JIRA starts sending events to Deskpro
- **URL**: The URL in your clipboard we copied from the Deskpro webhook
- **Issue Related Events**: “Comment > created“ - we’re only interested in newly created comments

Click the “Create“ button at the bottom of the webhook screen. That’s it - we’ve created our webhook in JIRA. Tickets associated with JIRA issues will now update their status to “Awaiting Agent“ when new comments are added to issues via JIRA itself.

System

WebHooks

New Webhook Listener

Deskpro Test Webhook

Test Apps v3 Webhook

Name *

Deskpro Status Change Webhook

Status *

Enabled | Disabled

URL *

https://mycompany.deskpro.com/api/v2/webhooks/CB8QAJ887TAL1B7X/invoc

You can use the following additional variables in the URL: ${attachment.id}, ${board.id}, ${comment.id}, ${issue.id}, ${issue.key}, ${mergedVersion.id}, ${modifiedUser.accountId}, ${modifiedUser.key}, ${modifiedUser.name}, ${project.id}, ${project.key}, ${property.key}, ${sprint.id}, ${version.id}, ${worklog.id}

Read more

Description

Events

Issue related events

You can specify a JQL query to send only events triggered by matching issues. The JQL filter does not apply to events under the Issue link column.

All issues

Syntax help

Attachment	Issue link	Issue	Worklog	Comment	Entity property
☐ created	☐ created	☐ created	☐ created	☑ created	☐ created or updated
☐ deleted	☐ deleted	☐ updated	☐ updated	☐ updated	☐ deleted
		☐ deleted	☐ deleted	☐ deleted	