Agile / Agile tutorials / Automatically escalate overdue issues in Jira

TUTORIAL

# Automatically escalate overdue issues in Jira



BY KEV ZETTLER

#### **BROWSE TOPICS**

Agile manifesto

> The agile advantage
DevOps

- > Agile Teams
- Agile tutorials

Overview

How to do scrum with Jira Software

Learn advanced scrum with

Learn kanban with Jira Software

Learn how to use Epics in Jira Software

Learn how to create an agile board in Jira Software

Learn how to use sprints in Jira Software

Learn Versions with Jira Software

Learn Issues with Jira Software

Learn burndown charts with Jira Software

Auto-create sub-tasks and update fields in Jira

How to automatically assign issues with Jira Software Automation

How to sync epics stories with Jira Software Automation

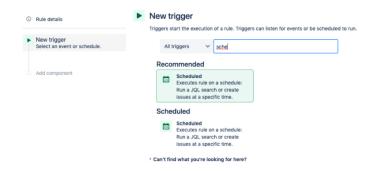
- Automatically escalate overdue issues in Jira
- > About the Agile Coach

All articles

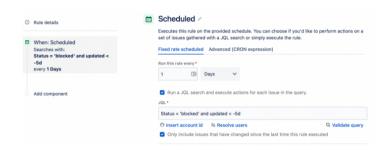
This guide will detail how to create a new automation rule that searches for overdue issues once a day, sends a Slack reminder to the team, and adds a comment to the issue. You will need an active Jira project.

### Step 1. Create a rule

Navigate to the Automation space in your project settings. In the Rule tab, click on the **Create rule button** in the top right corner of the screen. On the **New Trigger** screen search for and select the **Scheduled** trigger

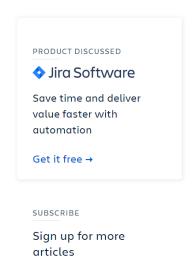


Configure the **Scheduled** action to query for issues with a "blocked" status that haven't been updated for more than five days. This uses JQL to dynamically select a set of issues.



#### Step 2. Add component

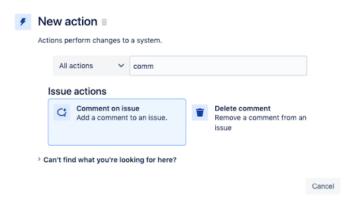
Select **Add component** from the Rule details list on the left sidebar Search for and select the **Commont** 



Email
email@example.com

Subscribe

on **issue** component.



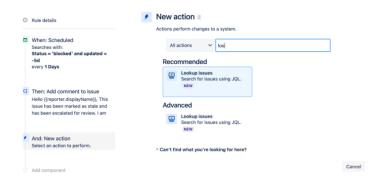
Use the **Comment on issue** screen to compose a comment message that the automation rule will automatically apply to transitioned issues. The comment acts as a template string and can interpolate data. The "How do I access issue data in my comment?" link provides examples of how to use the templating system.

Below the comment is configured to inject the **reporter.displayName** value. You can customize your comment to your liking. Click the **Save** button when completed.

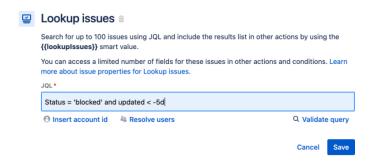


Step 3. Add a Lookup issues action

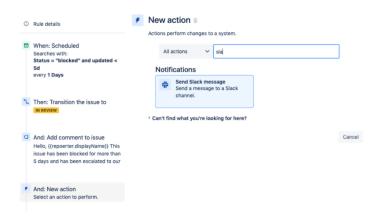
Next, add **New action** and search for the **Lookup** issues action.



This action aggregates the queried issues into a list for batch actions. This is useful to avoid duplicate follow-up actions. Configure the **Lookup issues** action so it uses the same JQL query as the schedule trigger.

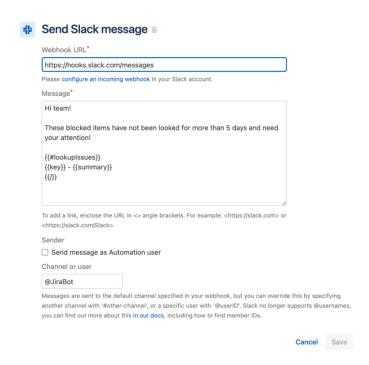


The following is another example of adding an action component to this rule. On the **Add Action** screen, search for and select the **Send Slack message** action.



## Step 4. Add Slack message

To enable the **Send Slack message** action you will need the Webhook URL, which you can get from your Slack administrator. This allows you to specify the message, channel, or user to send the Slack message to. Once you configure the **Send Slack message** action, click the **Save** button.



**Rule details** on the left sidebar, it should look like the following:

- Rule details
- When: Scheduled
  Searches with:
  Status = 'blocked' and updated <
  -5d
  every 1 Days
- Then: Add comment to issue
  Hello {{reporter.displayName}}, This
  issue has been marked as stale and
  has been escalated for review. I am