Integration Manual

Jira

How to integrate cloud-based Jira implementations

Document Information

Code: **IM-JIRA**
Version: **2.0**
Date: **17 April 2025**

Admin By Request
ZERO TRUST PLATFORM

Copyright © 2025 Admin By Request

Contact Admin By Request

+64 21 023 57020
marketing@adminbyrequest.com
adminbyrequest.com
Unit C, 21-23 Elliot St, Papakura, NZ

Table of Contents

Overview

Introduction

This topic describes integration between Admin By Request and Jira for cloud-based Jira implementations. The integration is relatively simple, comprising three mechanisms: one to create issues in Jira from requests made in ABR, one to handle approved requests and one to handle declined requests. This is the extent of our Jira integration; it is up to the customer to further automate the updating of tickets through Jira automation rules.

Nevertheless, we include in this topic two simple automation rules to illustrate how some things might be done. Customers can use these as the basis for their own automation workflows.

IMPORTANT

- The integration covered here is available for Jira Cloud, but not on-premise installations like Jira Data Center Edition.
- Disclaimer - Admin By Request is not a Jira consultancy and we do not have extensive experience working with Atlassian tools, including Jira. Our integrations and examples are provided for the customer's convenience; they do not carry any warranty whatsoever and must be used entirely at the customer's own risk. We recommend testing everything in a non-production environment and we are not liable for any downtime or loss of productivity or data that might result from using the integrations.

In this document

Prerequisites

To complete the setup, you will need the following:

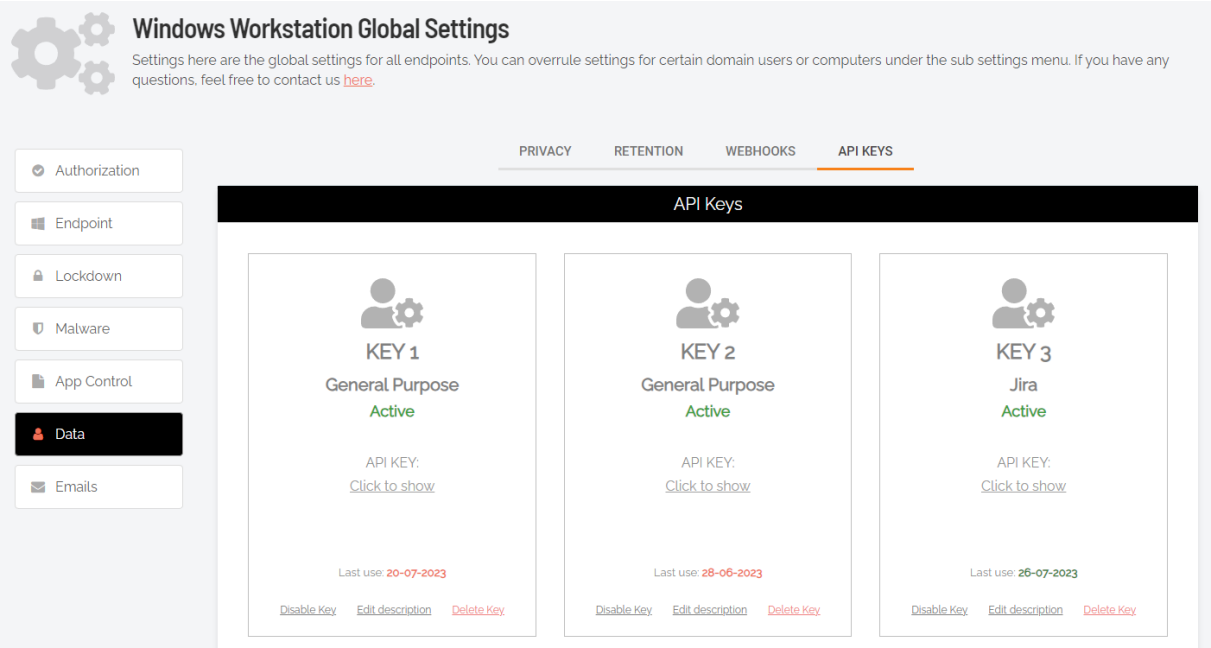
A. A Jira Service Management username and an Admin By Request Portal username.
You must have administrator access to both Jira Service Management and the ABR Portal.

B. A Jira Service Management API token associated with your username.
For information on managing API tokens in Jira, refer to https://support.atlassian.com/atlassian-account/docs/manage-api-tokens-for-your-atlassian-account/.

C. An API key in your ABR Portal:

Create an API key

1. Log in to the portal and navigate to **Settings > Tenant Settings > API Keys > API KEYS**.
2. To create a new API key, click button **Add New**.
3. If you want to name this key, click **Edit description** and name it accordingly.
4. Click **Save** to save the new API key:



Copy the key to the clipboard, ready for pasting in the next step.

Test the API key

The following test uses VS Code to send a curl statement to the ABR server's API interface.
A successful test returns **200 OK**, plus a list of ABR requests in JSON format.

> **NOTE**
> If there are no ABR requests in *any* category (pending, approved, denied or quarantined) then the server will still return **200 OK**, but no JSON data.

To test the API Key::

1. In your portal, identify the URL to use for API requests. This is based on the data center to which you are connected.

   To determine your data center, go to page Tenant Settings > API Keys in the portal and check which API prefix is shown under **About API Keys**. The data center (which is also the API prefix) will be one of the following:

   - **https://dc1api.adminbyrequest.com** (Europe)
   - **https://dc2api.adminbyrequest.com** (USA)

   Make a note of your prefix - among other things, this is the domain used when an API Key is created.

   > You can also see your API prefix on the API web pages (e.g. Public API > Auditlog API). However, a small script runs in the background that determines to which data center you are attached, so JavaScript must be enabled in your browser for this to work.

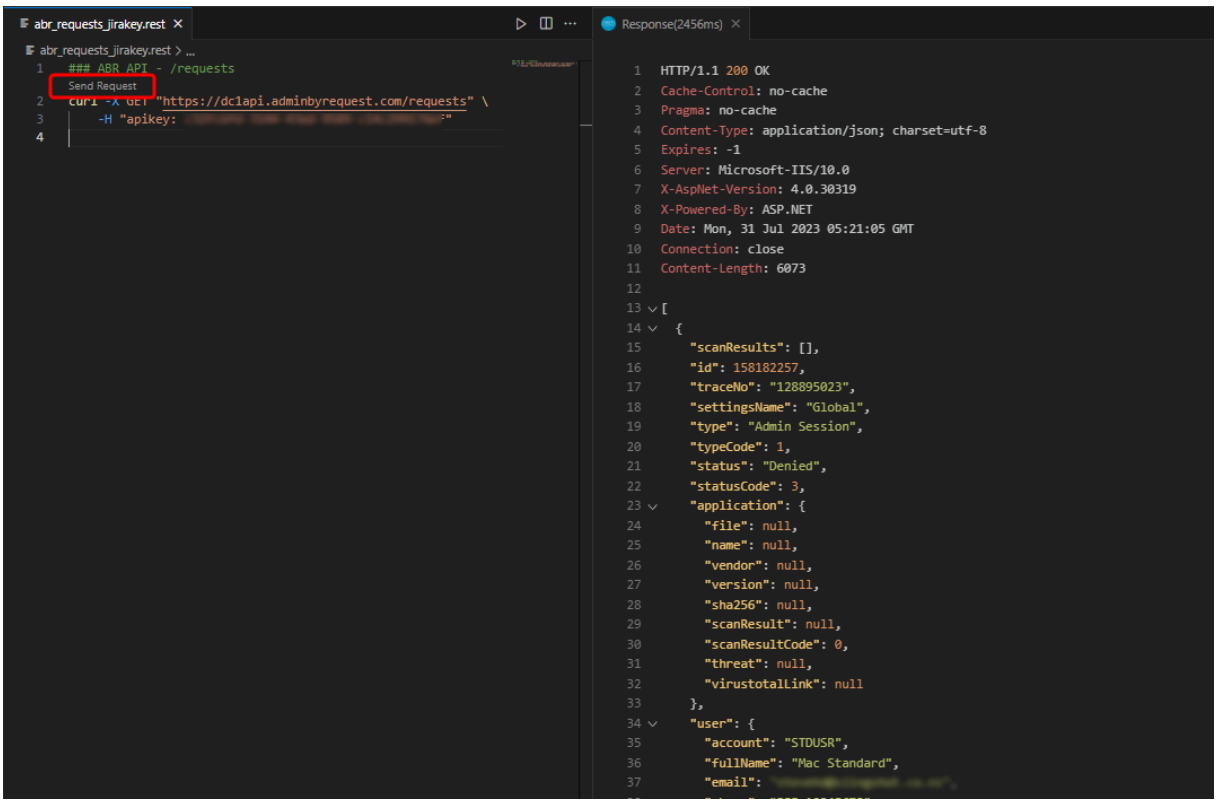
   The example below uses **https://dc1api.adminbyrequest.com/requests** as the URL.

2. Enter the following curl statement, using your URL and your API key:

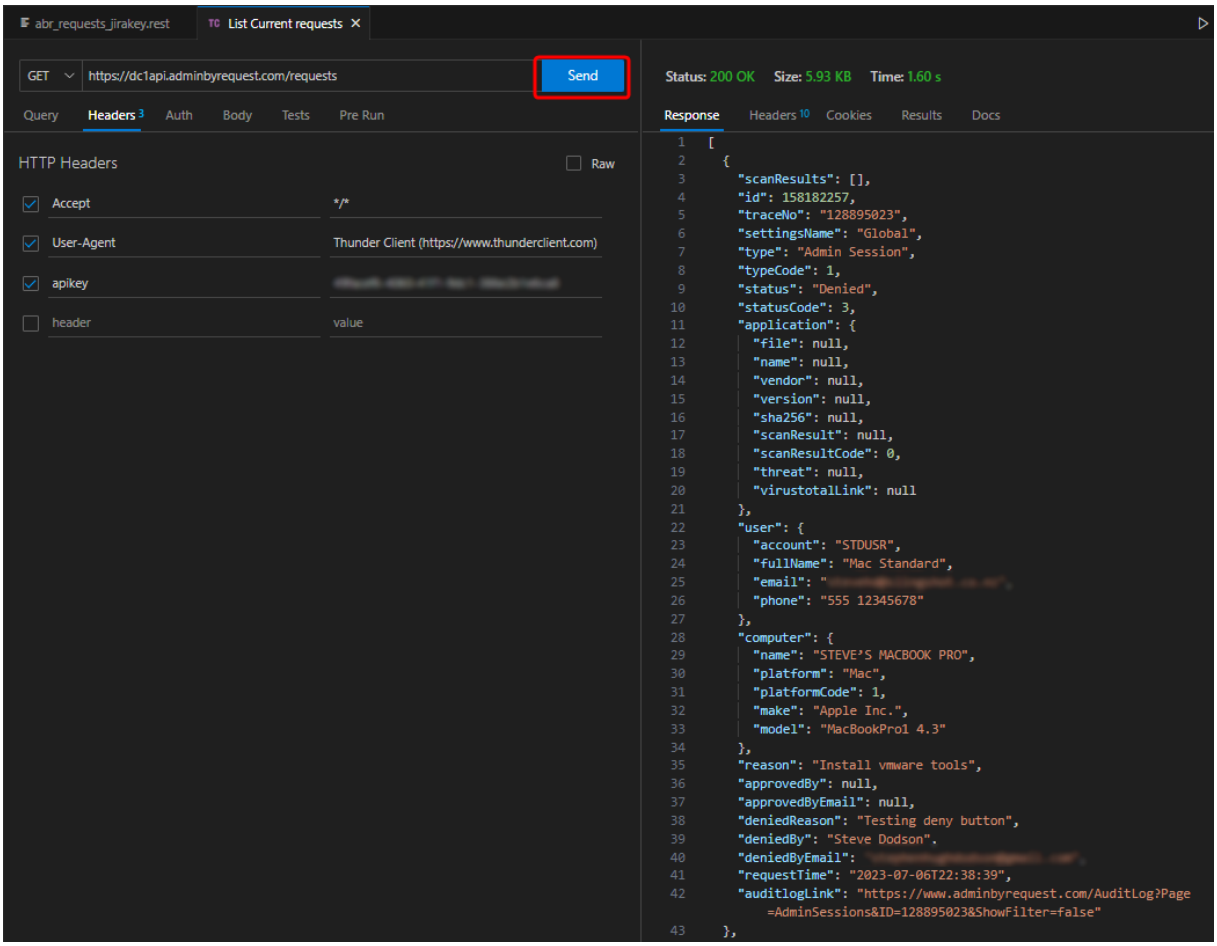
```
curl -X GET "https://dc1api.adminbyrequest.com/requests" \
    -H "apikey: <your API key>"
```

3. Send the curl statement. The following examples show the response in VS Code from the ABR server using firstly the REST client and secondly the Thunder client.

   REST client



Thunder client



There are multiple ABR requests returned across all states (pending, approved, denied, etc.) - only the first request is shown. Scroll the response you get back to see all requests.

Once you have this information, we're ready to get started.

There are three main tasks required to configure Jira integration. The next chapter describes each task in detail.

Integration Tasks

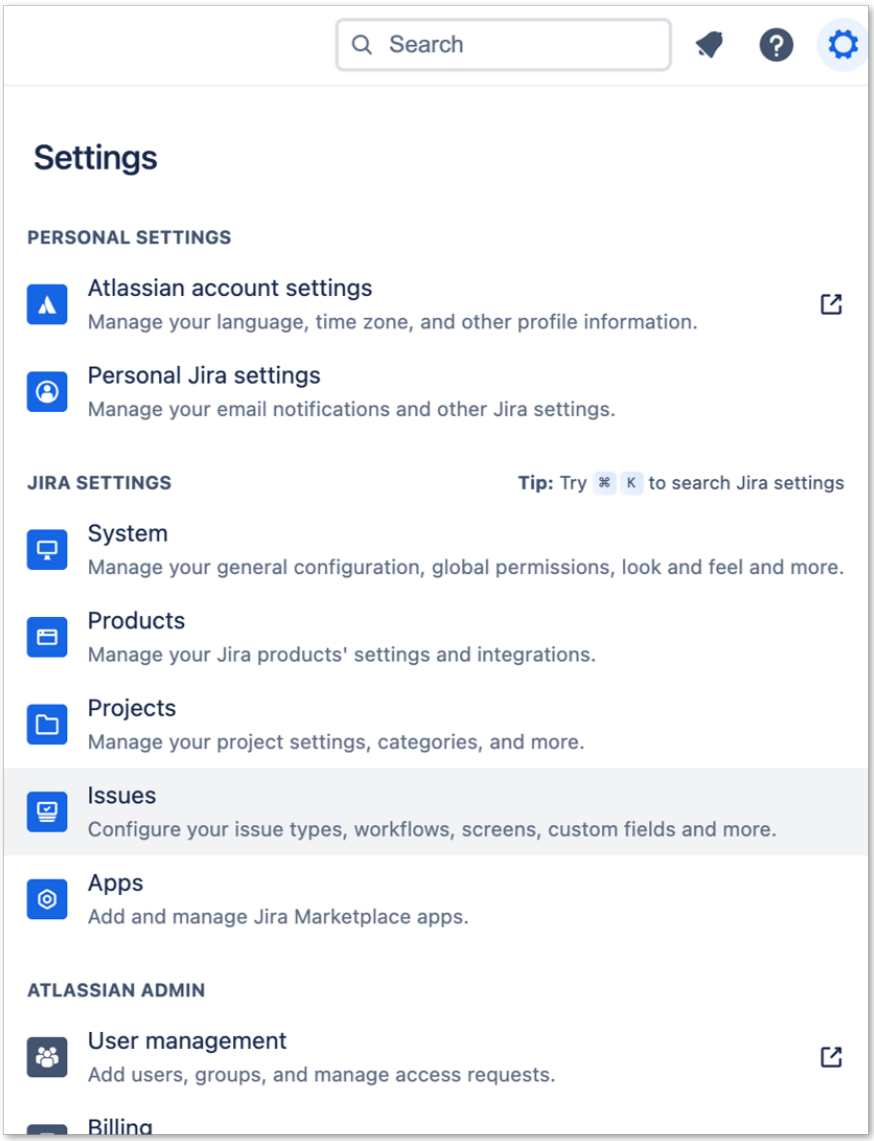
1. Creating custom fields

The integration requires you to set up a few custom fields. These fields hold information about:

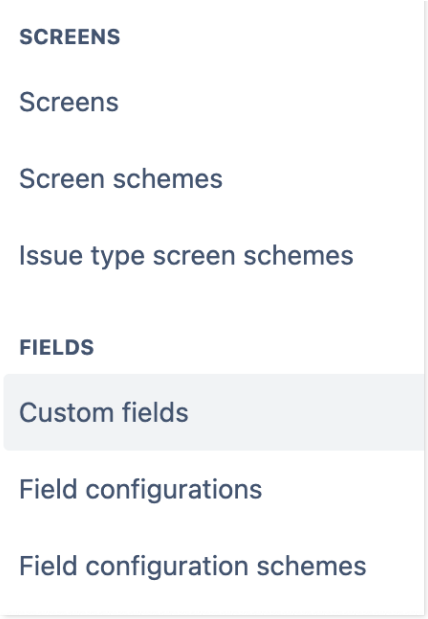
- The ID of the request from Admin By Request (can be used for further automation).
- The name of the approver (if the request is approved from the platform or via another integration).
- The reason supplied if a request is denied.

Create custom fields

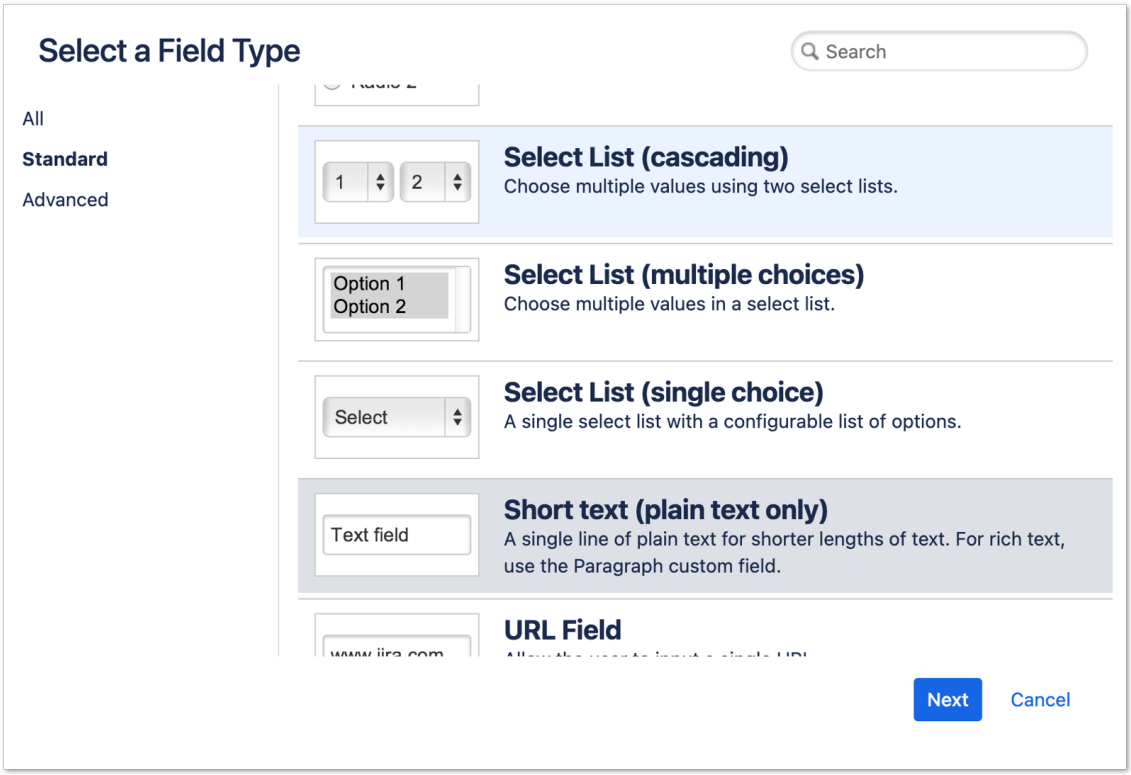
1. Log in to Jira Service Management and click the cog in the upper right corner to navigate to **Settings > Issues**:



2. In the left menu, select **Custom fields**:



3. In the *Custom fields* screen, click button **Create custom field** and select Field Type **Short text (plain text only)**:



4. Name the field (e.g. **ABR Request ID**) and, optionally, provide a description.
5. Repeat these steps for field **ABR Handled by**.

6. Repeat the steps one more time for field **ABR Reason**, but select the multiline option **Paragraph (supports rich text)** as the Field Type.

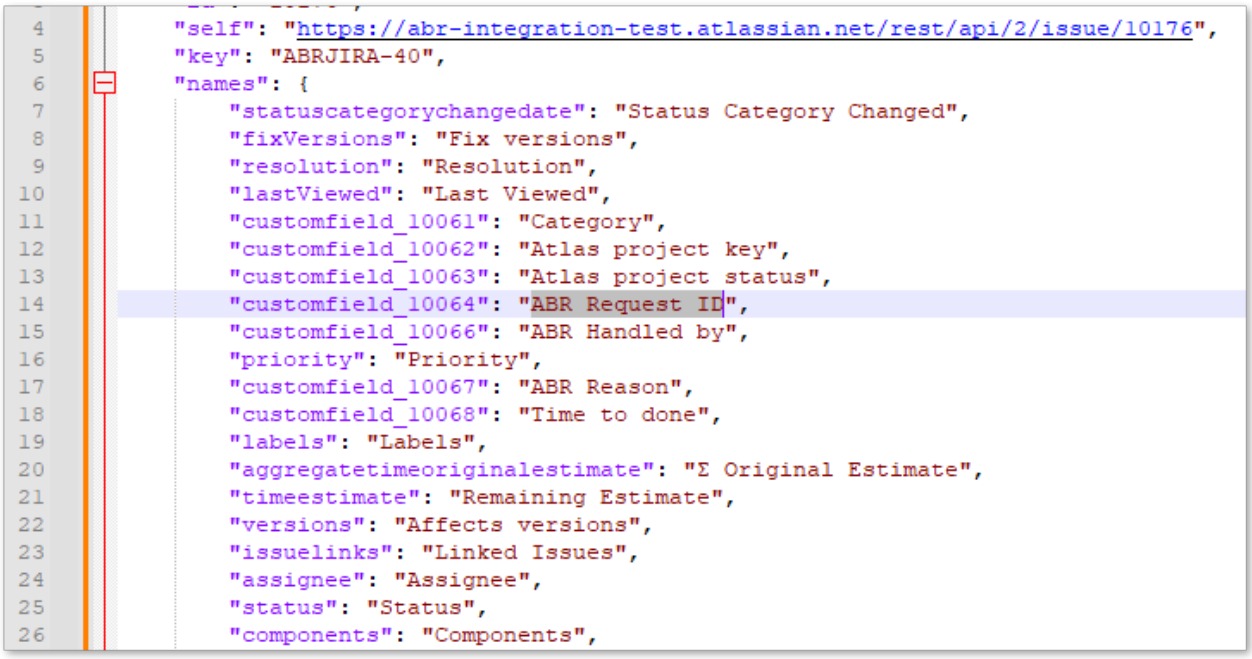
> **NOTE**
> You don't have to select any specific views for the fields - these will be assigned when creating the *Request Type* in the next section.

7. Finally, find the ids of the fields just created. To do this, make sure at least one issue exists in your project. Identify the issue number and enter the following URL in a browser:

```
https://<your-jira-website>.atlassian.net/rest/api/2/issue/<your-jira-issue-number>?expand=names
```

Copy and paste the resulting JSON into an editor that can format it nicely (e.g. Notepad++) and find the labels of the fields created.

The following example has issue number **ABRJIRA-40** (from the "key" field) and shows both field id (**customfield_10064**) and field name for **ABR Request ID**:



```
4   "self": "https://abr-integration-test.atlassian.net/rest/api/2/issue/10176",
5   "key": "ABRJIRA-40",
6   "names": {
7       "statuscategorychangedate": "Status Category Changed",
8       "fixVersions": "Fix versions",
9       "resolution": "Resolution",
10      "lastViewed": "Last Viewed",
11      "customfield_10061": "Category",
12      "customfield_10062": "Atlas project key",
13      "customfield_10063": "Atlas project status",
14      "customfield_10064": "ABR Request ID",
15      "customfield_10066": "ABR Handled by",
16      "priority": "Priority",
17      "customfield_10067": "ABR Reason",
18      "customfield_10068": "Time to done",
19      "labels": "Labels",
20      "aggregatetimeoriginalestimate": "Σ Original Estimate",
21      "timeestimate": "Remaining Estimate",
22      "versions": "Affects versions",
23      "issuelinks": "Linked Issues",
24      "assignee": "Assignee",
25      "status": "Status",
26      "components": "Components",
```

The field id for ABR Request ID is needed later in task "Overview" on page 1.

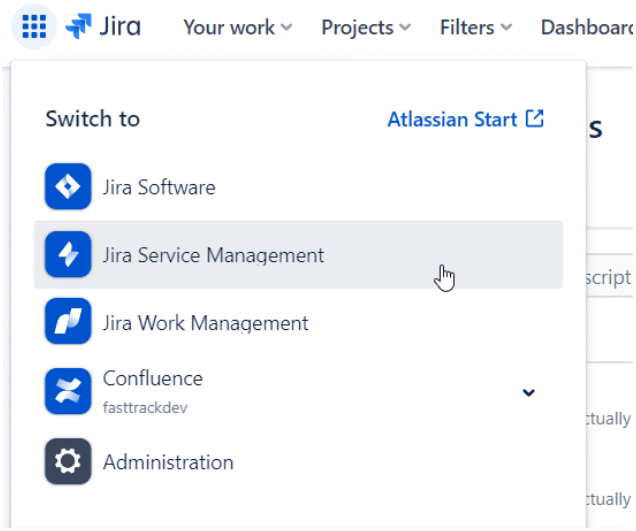
> **NOTE**
> This is an example - your field id(s) will almost certainly be different.

2. Creating a custom Request Type

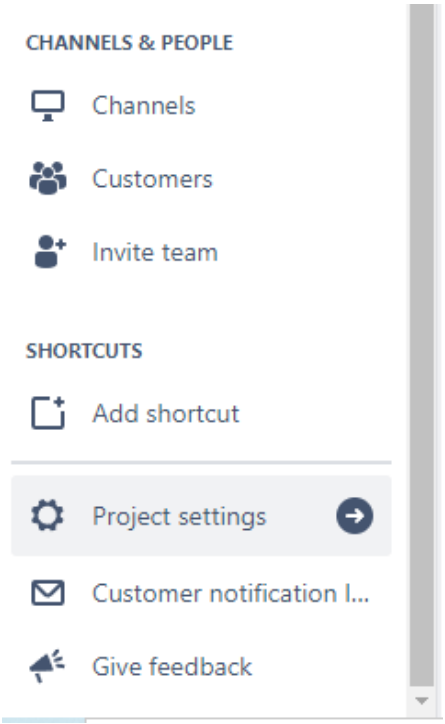
A custom Request Type enables Jira to identify any requests coming from theAdmin By Request servers.

Create a custom Request Type

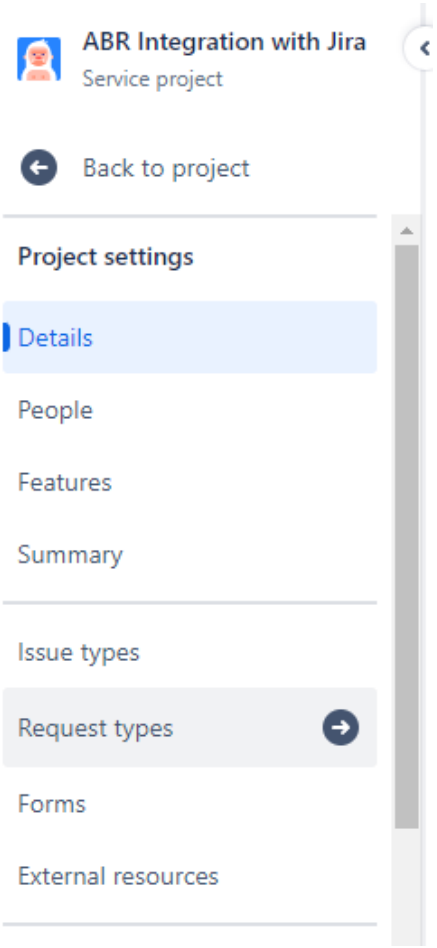
1. Login to Jira and, from the Switch to... menu, select Jira Service Management:



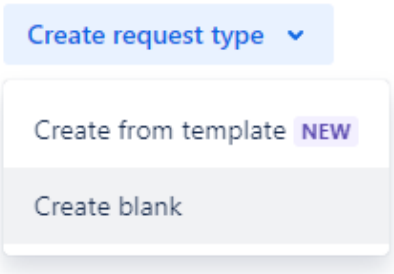
2. From within Jira Service Management, either create a new project or select the project you want to integrate with. Open the project and select **Project settings** towards the bottom of the left menu:



3. From the Project settings menu, select **Request types**:



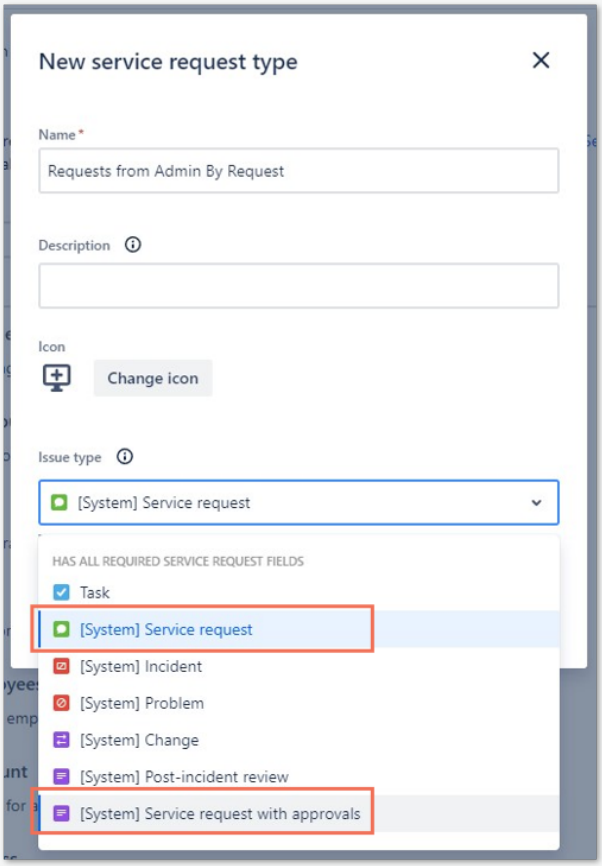
4. On the *Service requests* screen, use the **Create request type** button (top right) to create a request type (click the button and select **Create blank**):



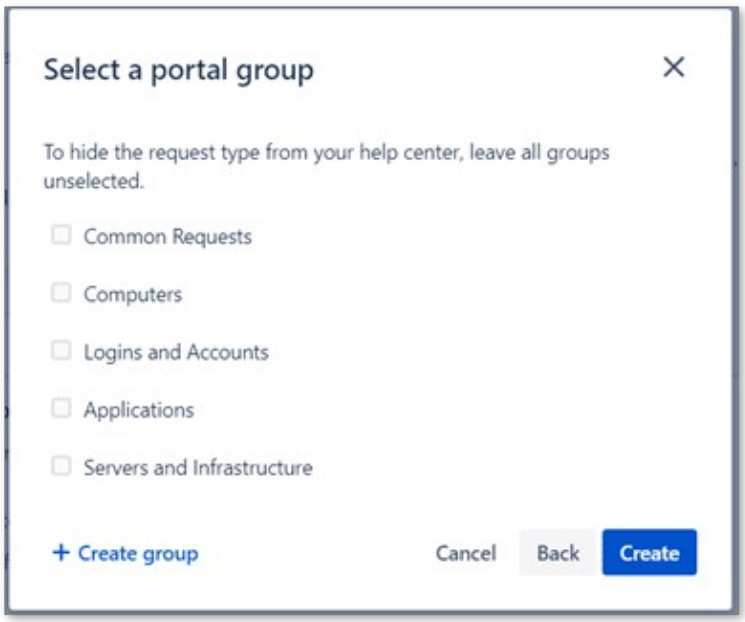
5. Give the request type a name and select the Issue type as either:
   - **[System] Service request**
     This sends requests to Jira and updates them as they get approved or denied, OR
   - **[System] Service request with approvals**
     Works in the same way as the ordinary service request type, but allows designated approvers to approve or deny requests from within Jira.
     With a bit of automation, this also enables Jira to call the Admin By RequestAPI to approve or deny requests directly from within Jira.

NOTE

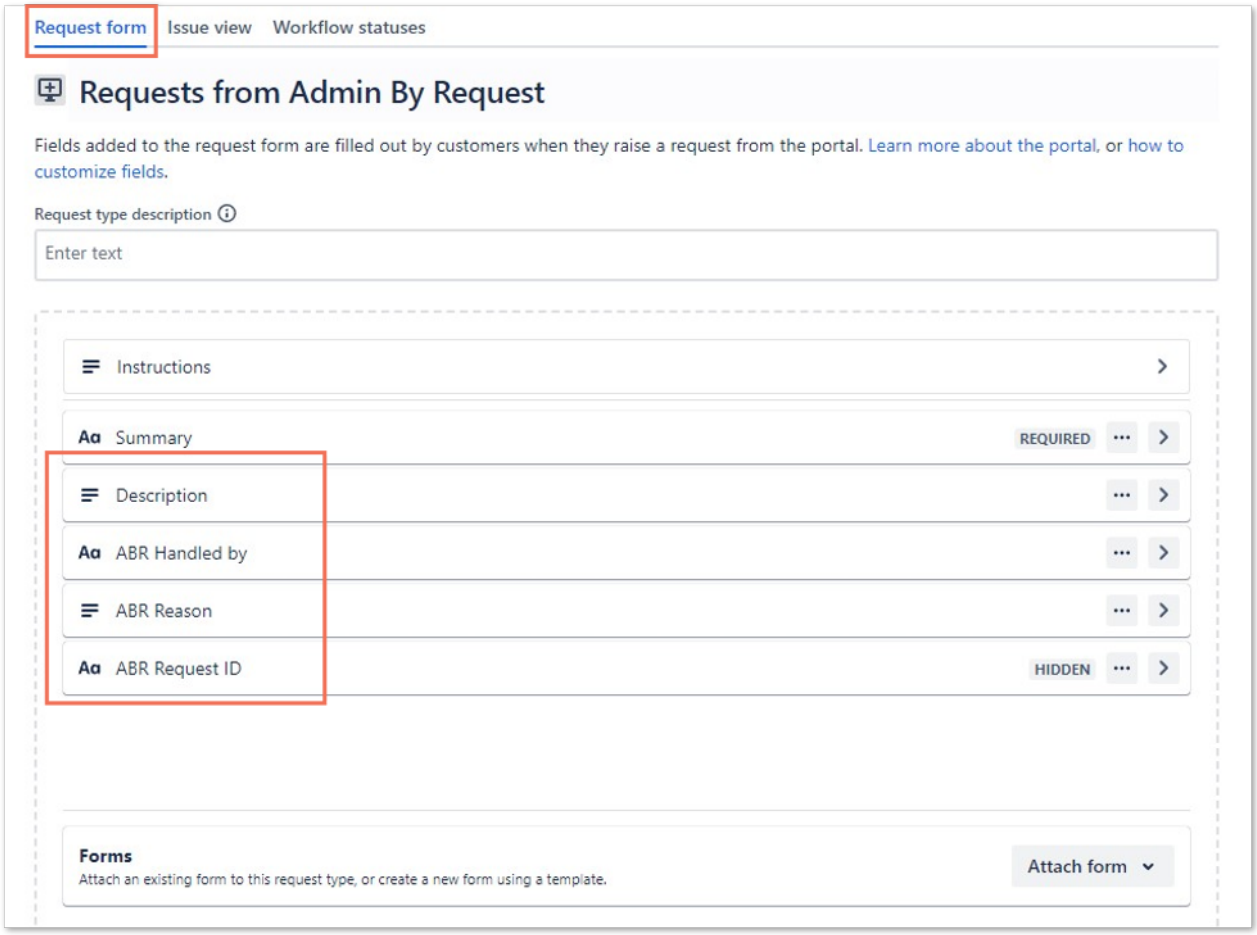
The automation example ("Overview" on page 1) requires that **[System] Service request with approvals** is the option chosen at this point:



6. Select the portal group to assign to this new request type (or leave all options blank):



7. Finally, drag and drop the three newly created custom fields, as well as the **Description** field into the view of the new request type:



NOTE

- Make sure you are on the **Request form** tab and not *Issue view* or *Workflow statuses*.
- Make sure the **Description** field is added along with the three custom fields. *Summary* and *Instructions* should already be there.
- You can leave the ABR Request ID field as "Use preset value and hide from portal" if you do not wish the ABR Request ID to be visible.

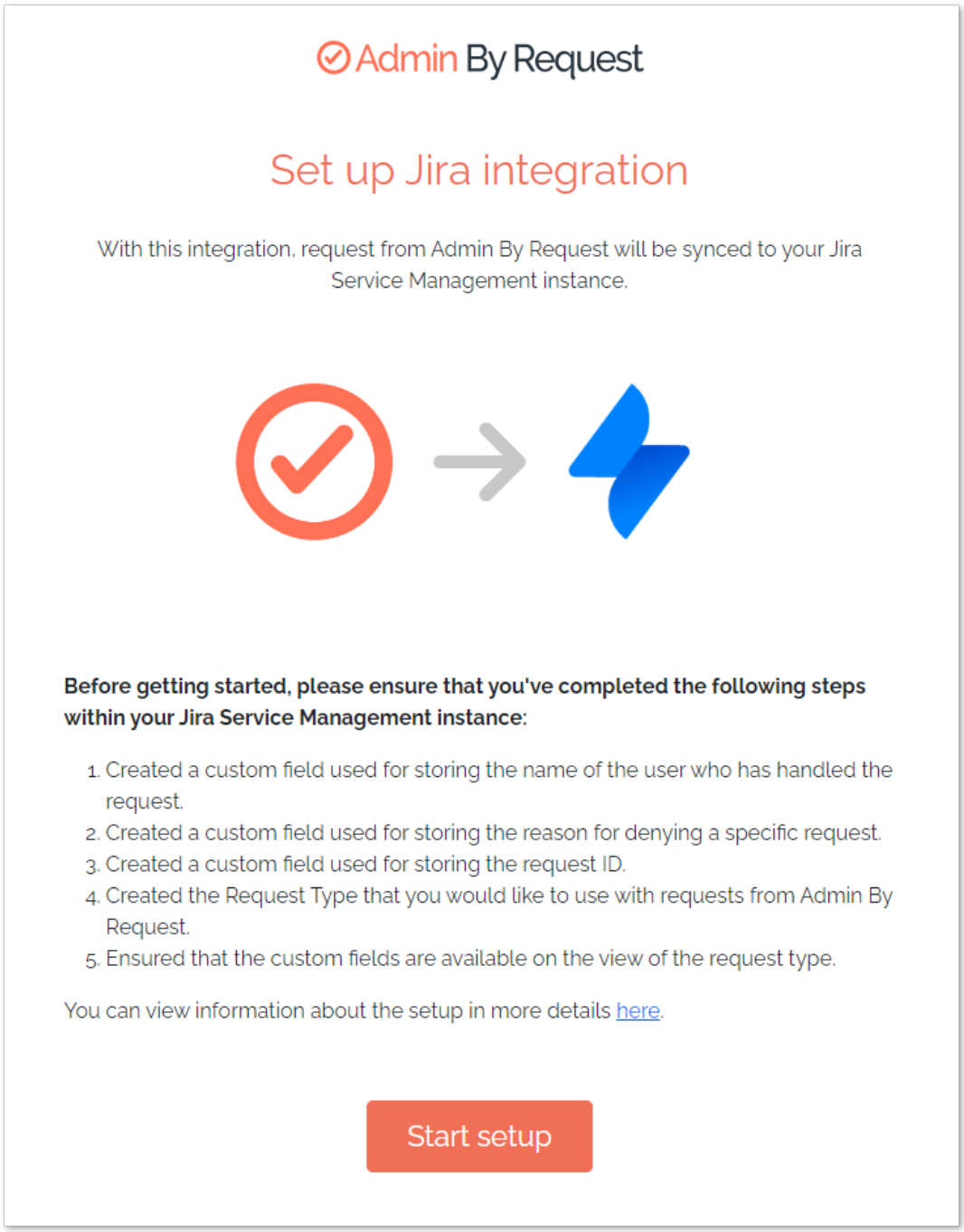
8. Save the changes.
   Everything is now ready to create the ABR-Jira integration.

3. Integrating Jira with Admin By Request

Now that the pre-requisites are in place, head over to https://jira.adminbyrequest.com to start the setup process.

Setup Jira integration

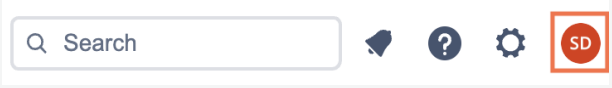
1. From the *Set up Jira integration* screen, check once again that you've completed all the prerequisite steps and click button **Start setup**:



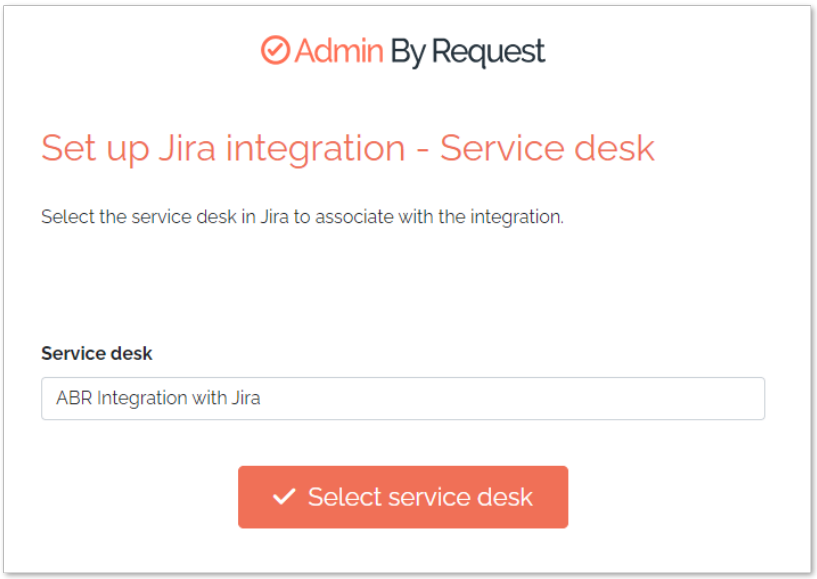
2. Retrieve your Atlassian API Token as well as your Admin By Request API key and enter these into the setup form alongside your Atlassian username and instance URL:

NOTE

Your username needs to be one that is authorized to make API calls. It is generally the email address recorded under your profile at the top right of the Jira Service Management screen:

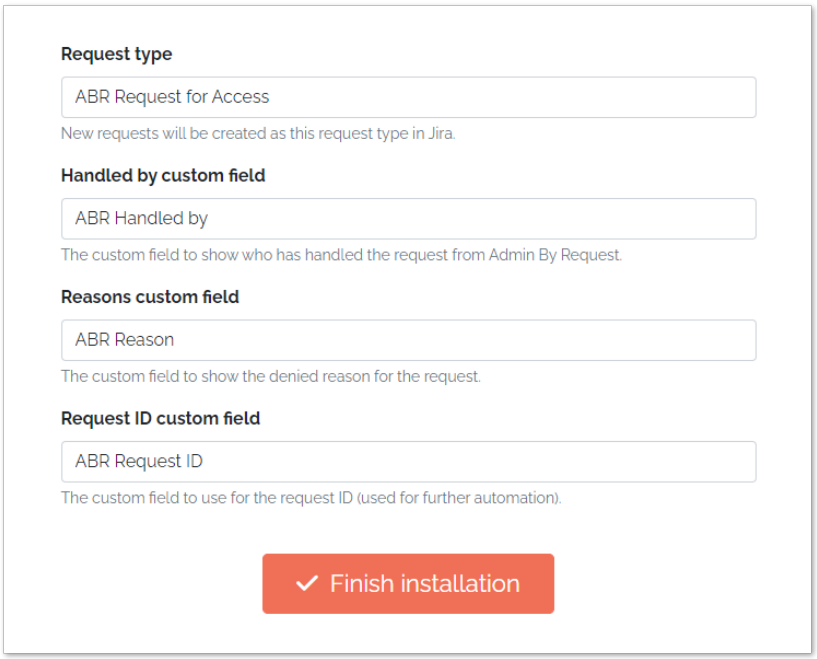


3. Next, you are prompted to select the Jira service desk to associate with the integration:



If you have more than one service desk project in Jira, a drop-down arrow will appear at the right of the *Service desk* field, allowing you to select the one you want.

4. The final screen is where the custom fields you created are associated with the three required by the ABR integration. *Request type* is already selected for you - choose the other three to match your custom fields:



5. Click **Finish installation** to complete setting up the Jira integration.
New requests from Admin By Request will now be sent to your Jira instance as service requests.

You can check the status of your Jira integration via the portal (**Settings > Integrations > Jira**):

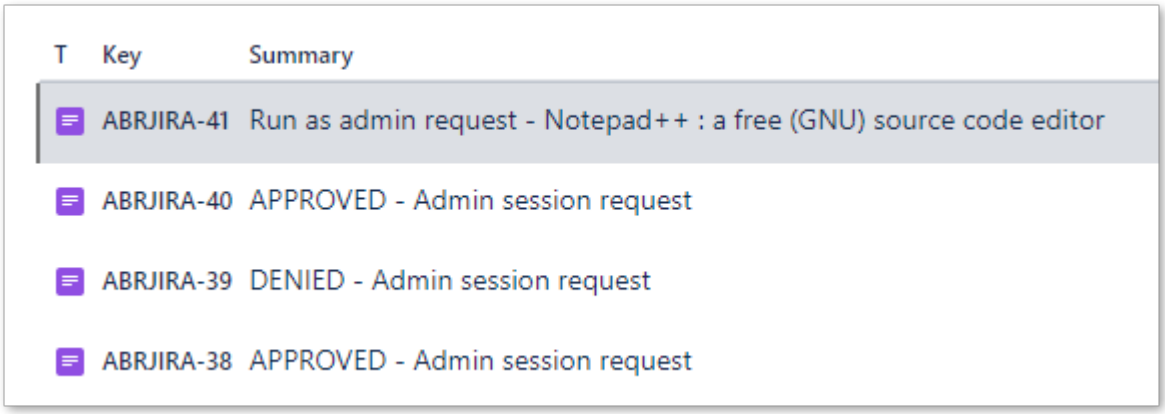
Updating Jira issues

When a user makes a request via ABR for either an "Admin Session" or to execute something "Run As Admin", a Jira issue is created in your selected service desk instance.

NOTE
It can take several minutes after the ABR request is submitted for the issue to be created in Jira.

The Jira *Summary* indicates the type of request: **Admin session** or **Run as admin**. This field is updated once the request is either APPROVED or DENIED.

For example, the following list of issues shows one "Run as admin request" waiting for approval, two "Admin session" requests approved and one admin session denied:



What next?

The preceding sections cover the extent of the Admin By Request Jira integration. If you want to do more with Jira issues once they have been created, you can use Jira automation to populate issues with more information or to carry out actions such as email notifications.

IMPORTANT
The examples provided in the next section are for convenience only - we do not support any Jira automation rules and they must be used entirely at the customer's own risk.

If you do decide to work with Jira automation, note the following:

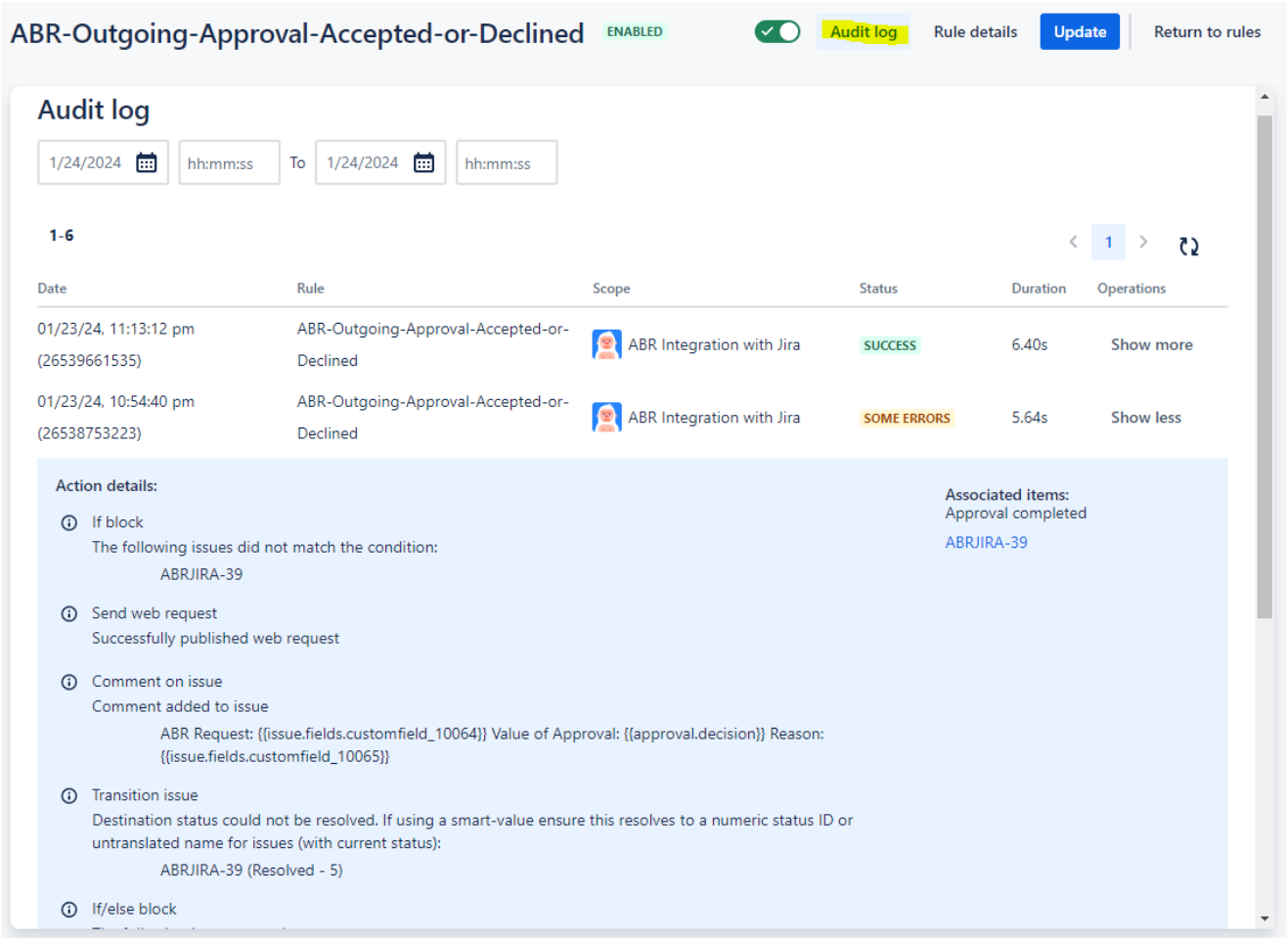
- If you selected **Service request** as the Request Type (see "2. Creating a custom Request Type" on page 8), you can use simple automation to interrogate the Summary field to carry out further actions (illustrated in "Example 1 - Populating issue fields after issue creation" on page 18).
- If you selected **Service request with Approval** as the Request Type, you can use more advanced automation, as illustrated in "Example 2 - Approving or Denying ABR Requests in Jira" on page 21.

Creating Jira automation rules

Using the Jira Audit log

The Jira Audit log is a very useful tool when working with Jira automation, especially as an aid in troubleshooting.

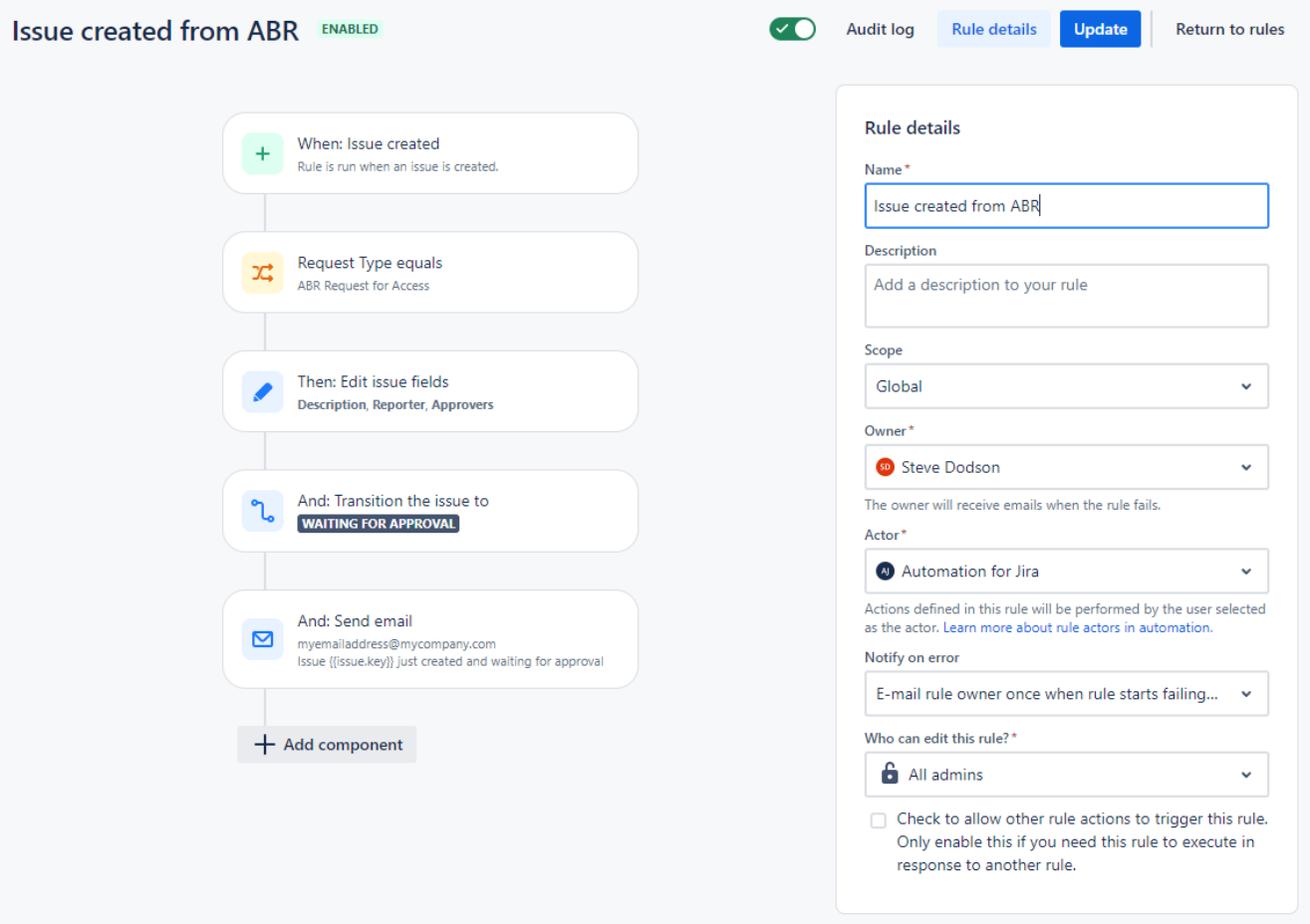
Each rule has its own Audit Log:



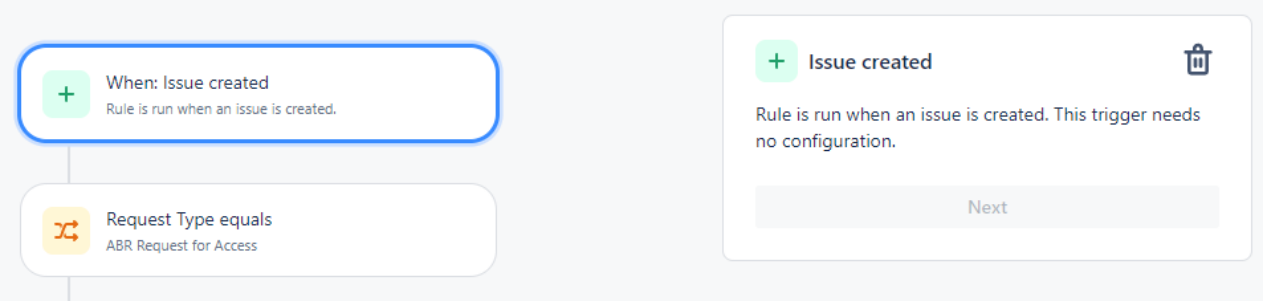
Linking back to the ABR Portal

A future release of the Jira integration will almost certainly include a link back to the ABR Portal "Requests" page, from where requests can be approved or denied. Until this is available, *Example 1* below provides a workaround that customers can implement using a Jira rule (Action 1 - set fields, Description).

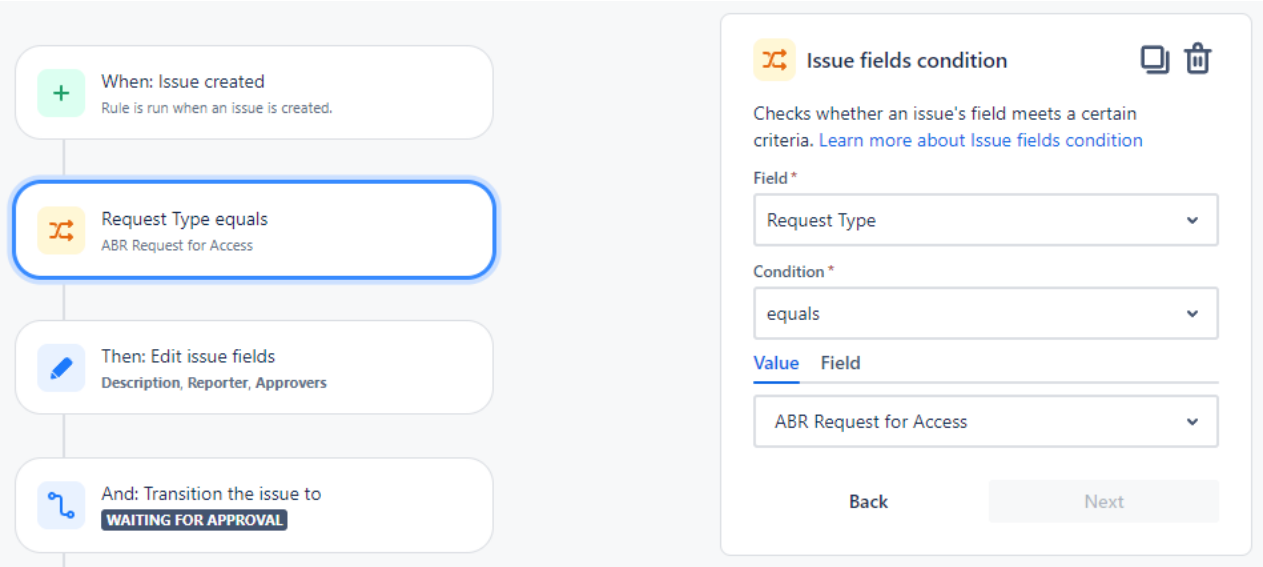
Example 1 - Populating issue fields after issue creation



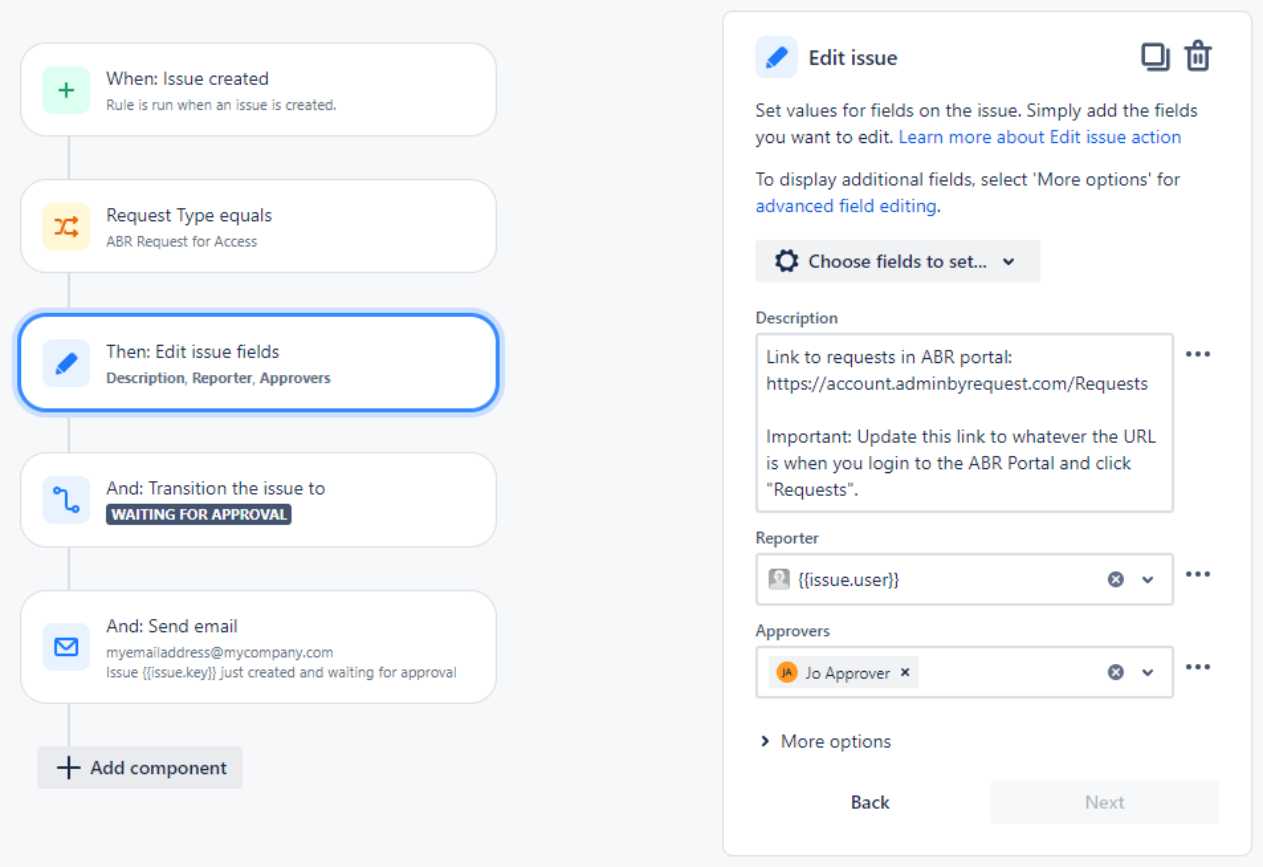
1. Trigger:



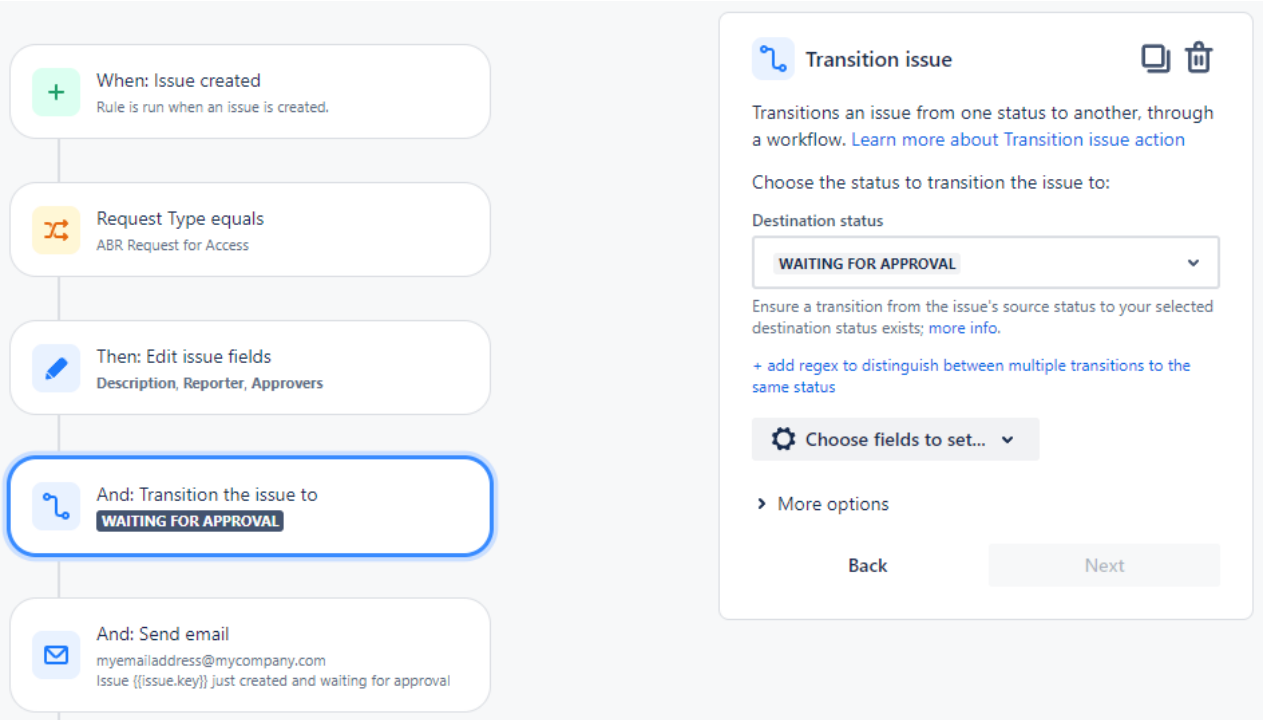
2. IF Request Type is **ABR Request for Access**:



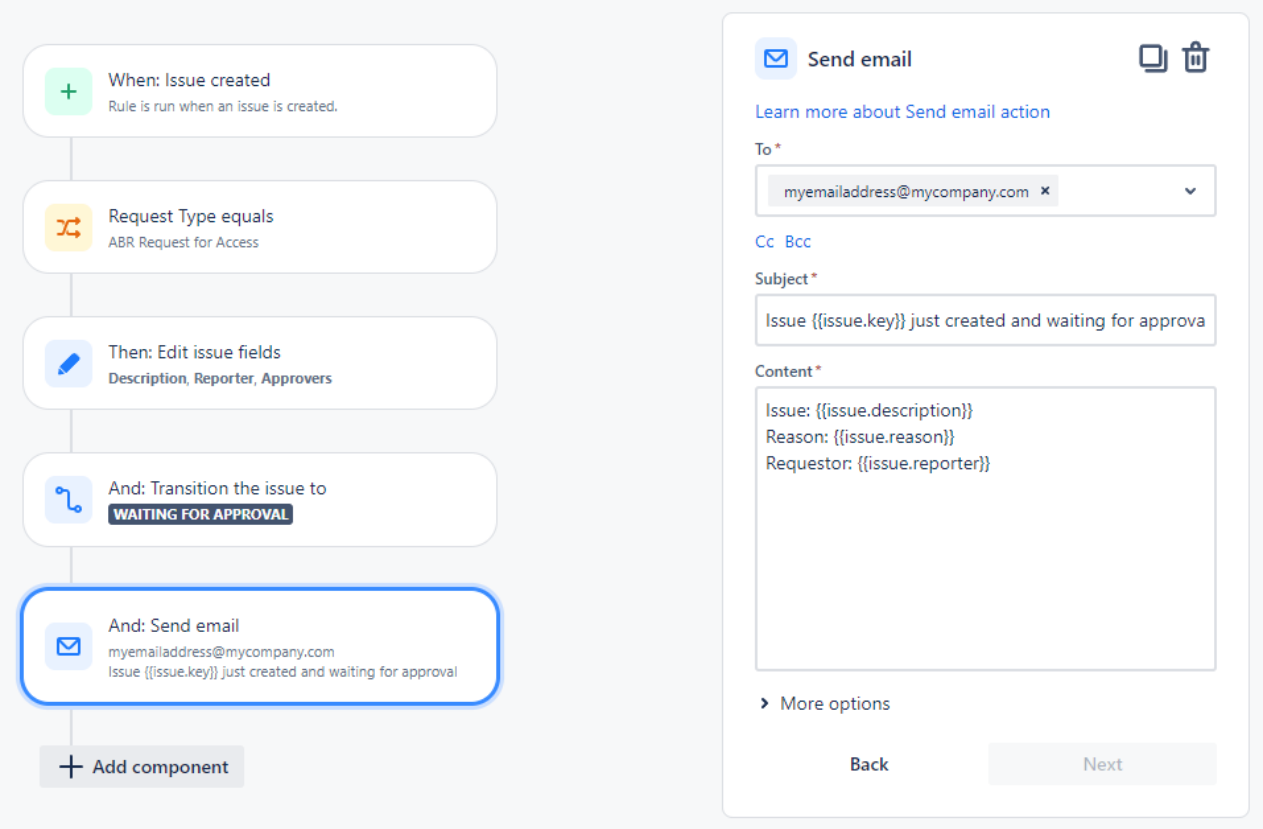
3. Action 1 - set fields:



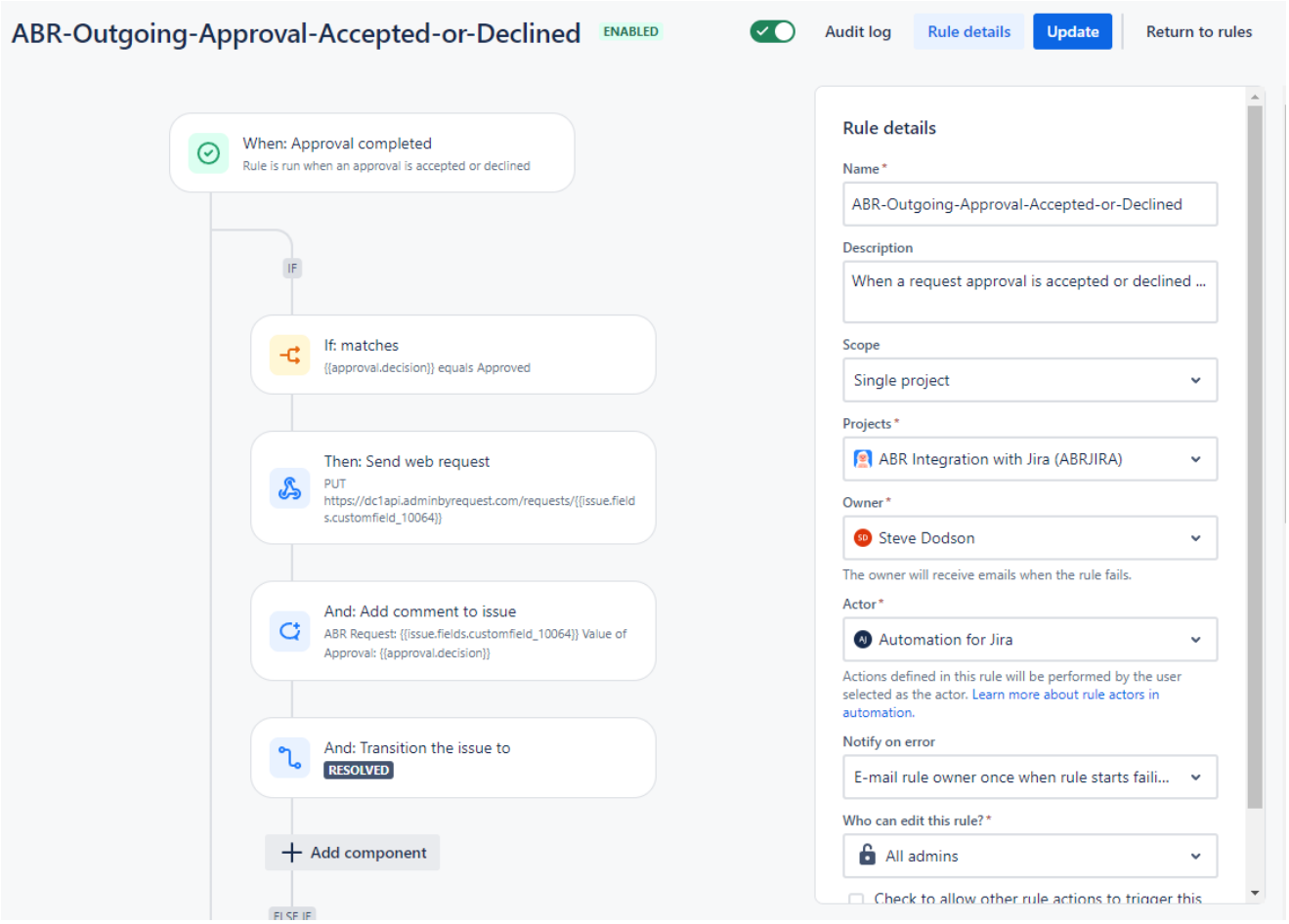
4. Action 2 - Set Status to **Waiting for Approval** (possibly optional, depending on workflow):



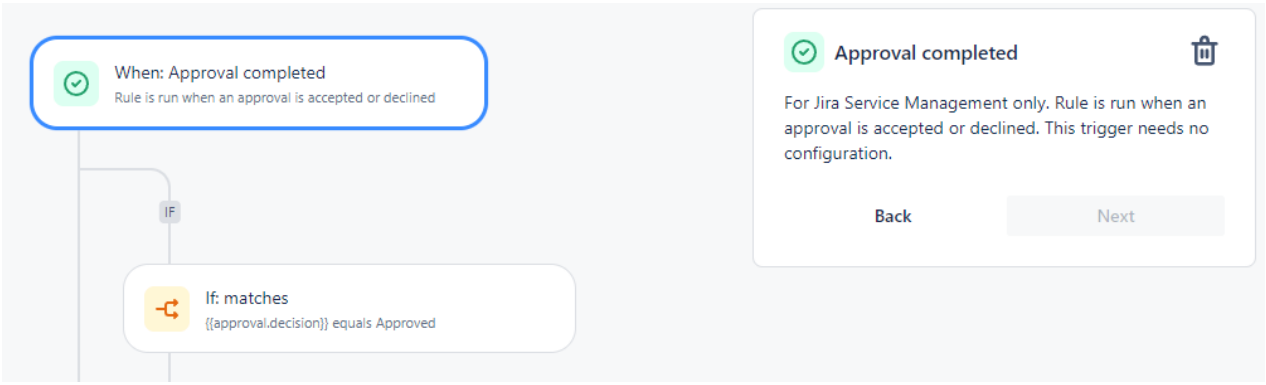
5. Action 3 - Send email:



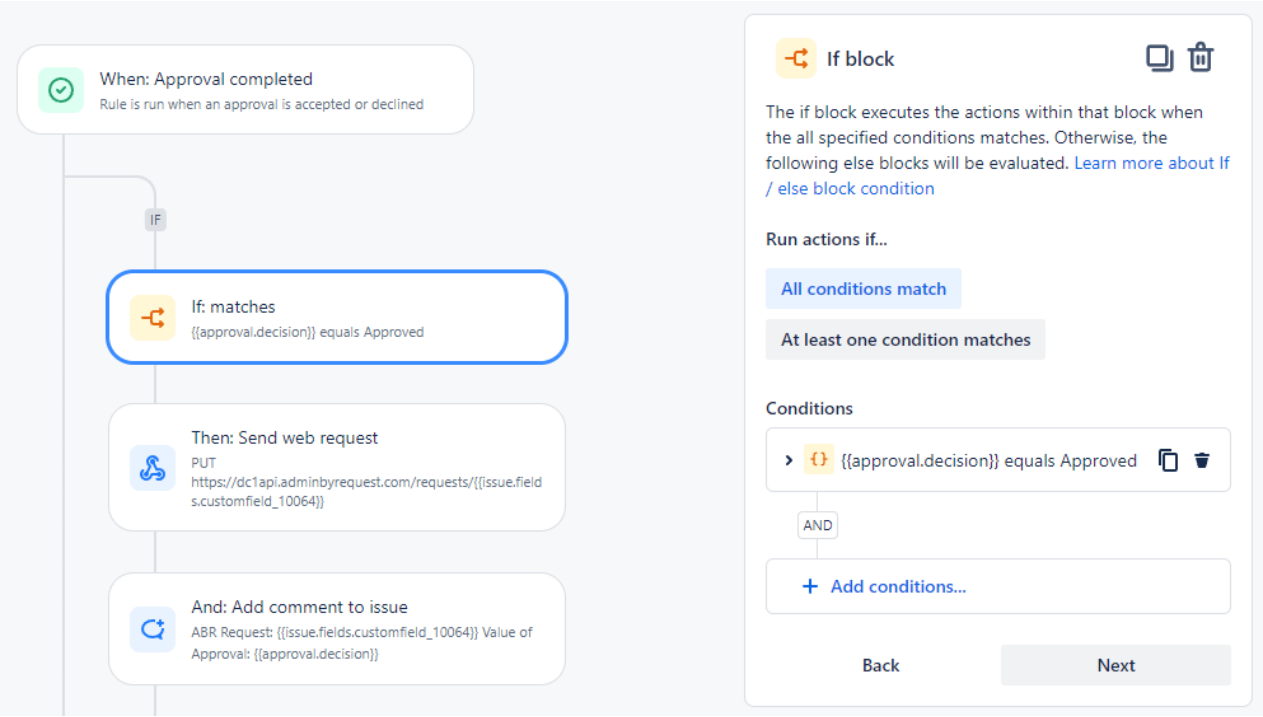
Example 2 - Approving or Denying ABR Requests in Jira



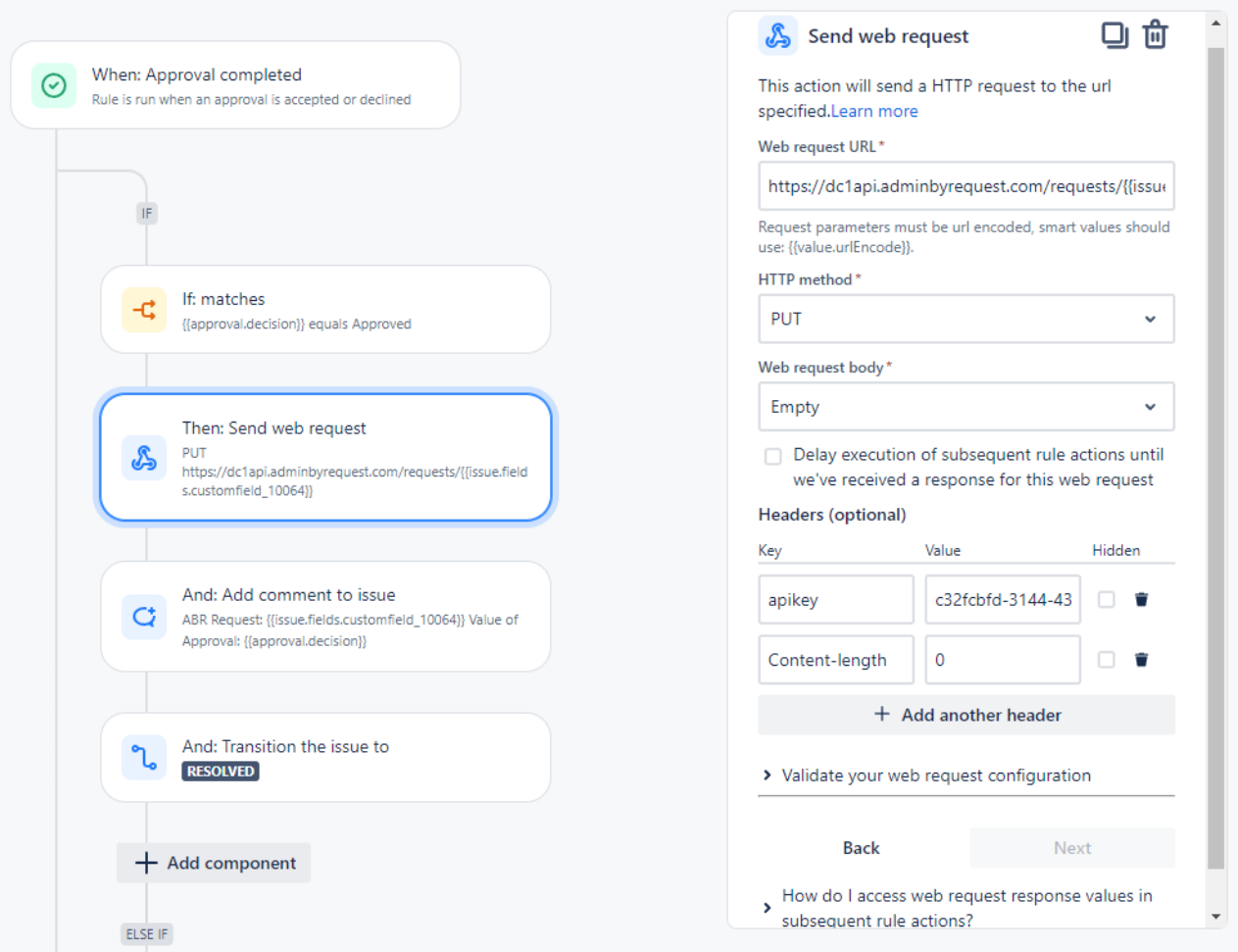
1. Trigger:



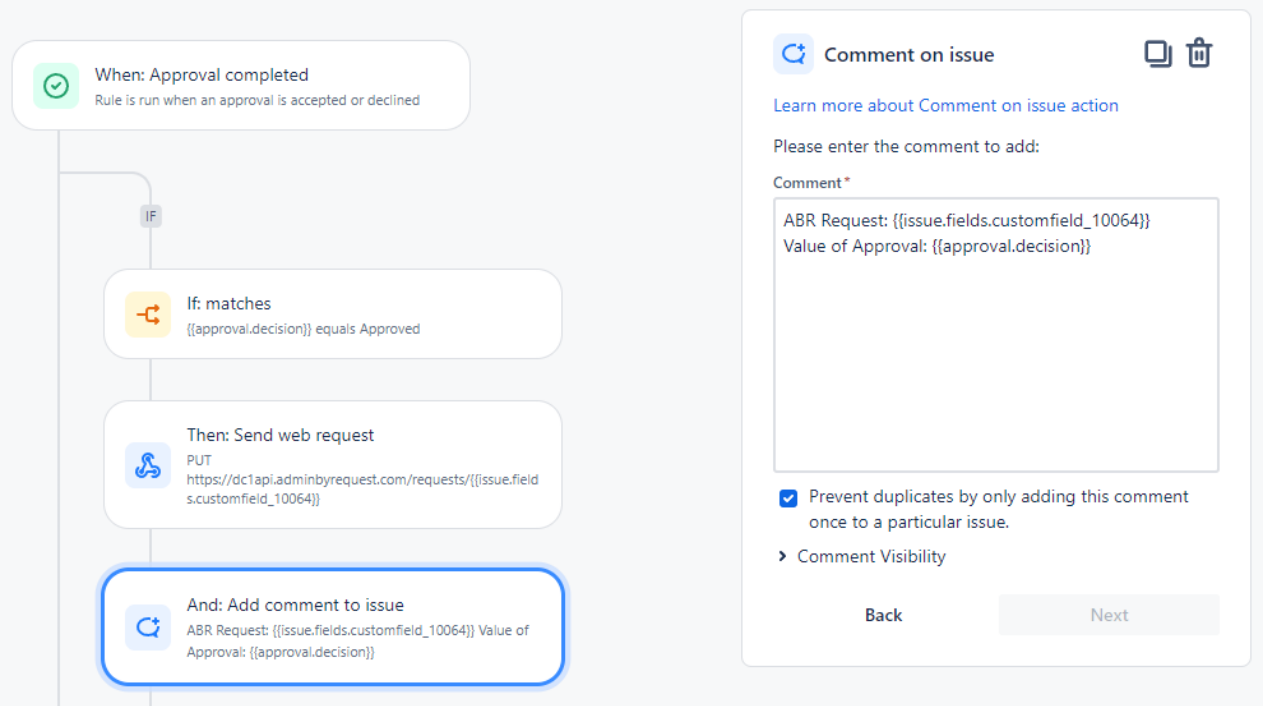
2. IF Approved:



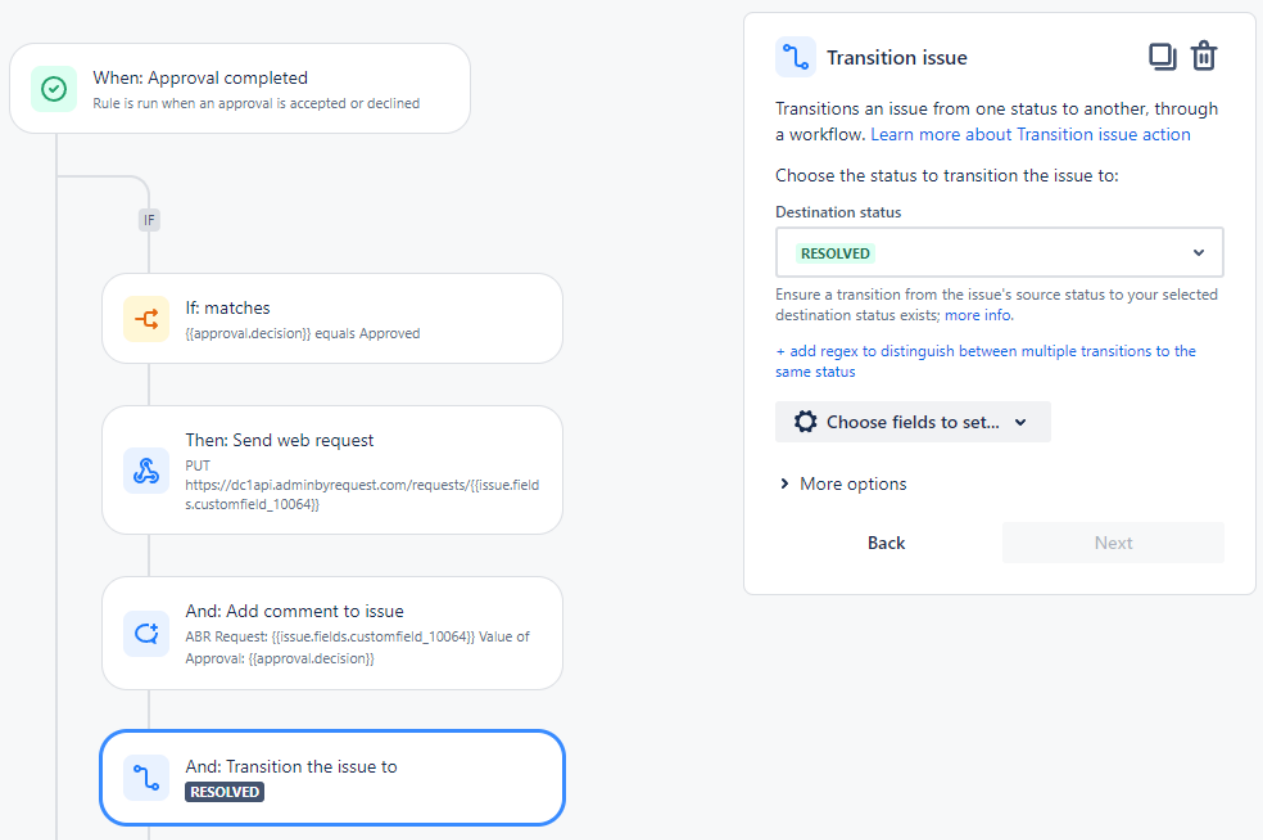
3. Action 1 - Send web request (PUT). Important - make sure you read up on API calls here (API Overview) and here (Requests API):



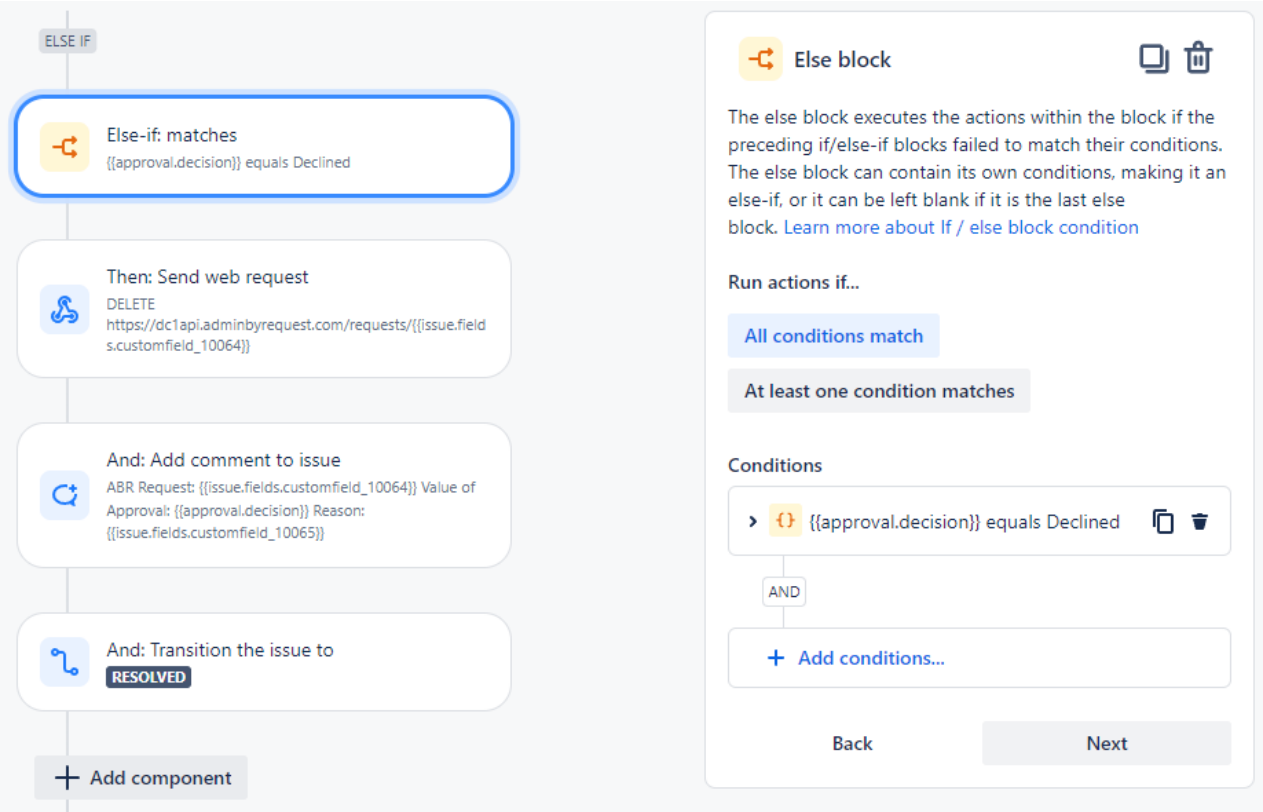
4. Action 2 - Add Comment to Jira issue (Important - make sure you identify the correct customfield ids - yours might be different from the example):



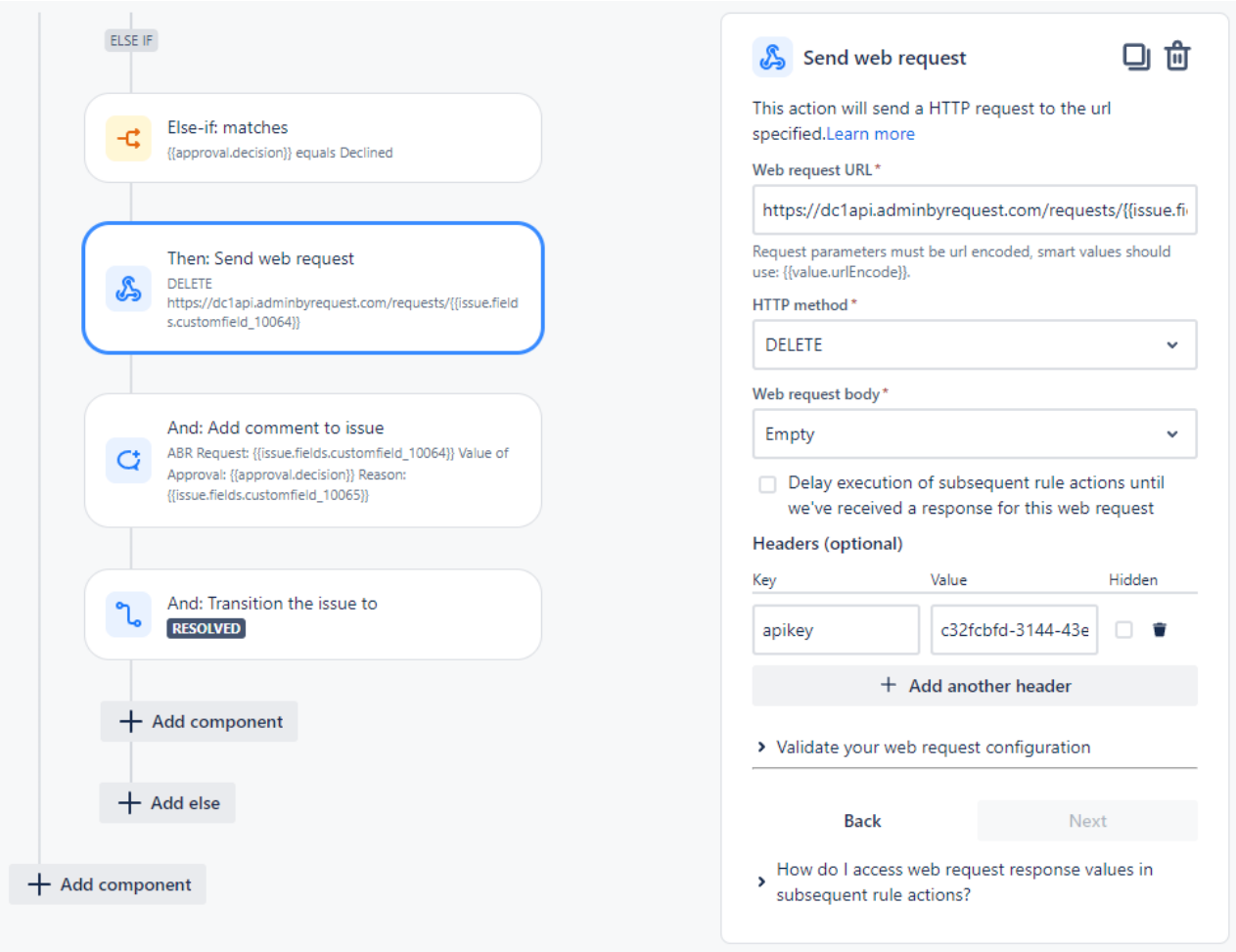
5. Action 3 - Set Status to **Resolved**:



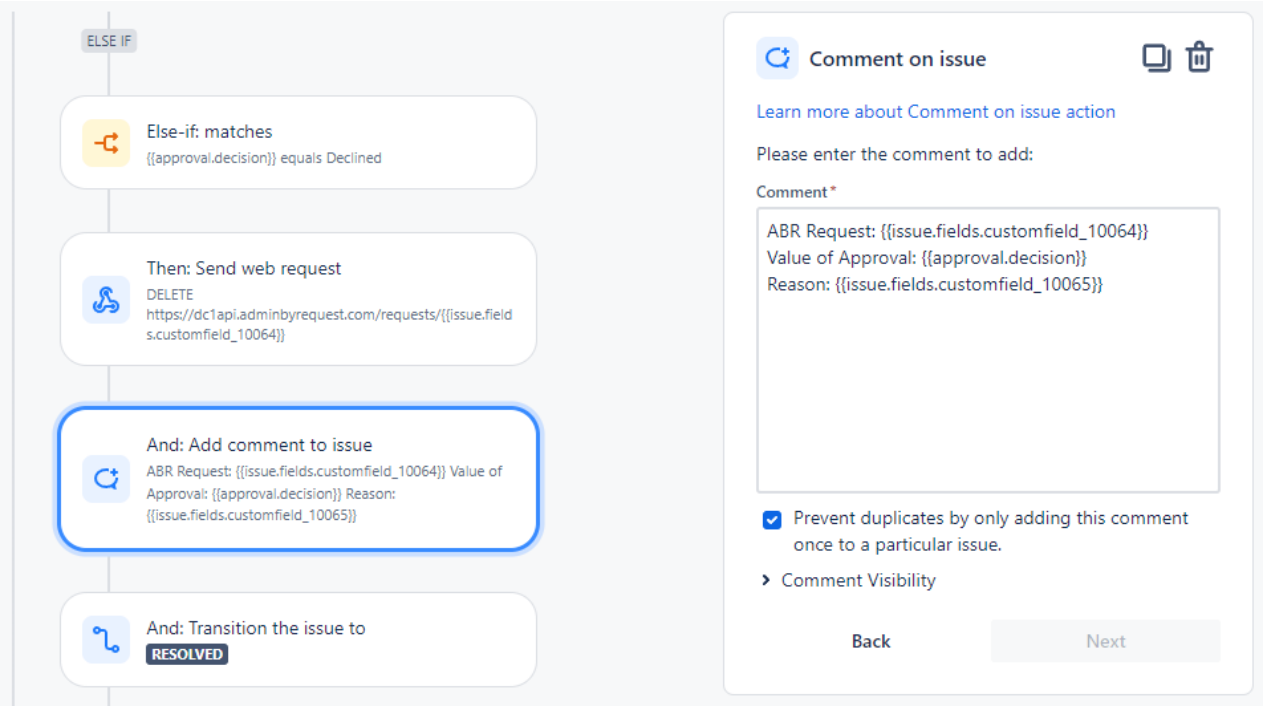
6. IF Declined:



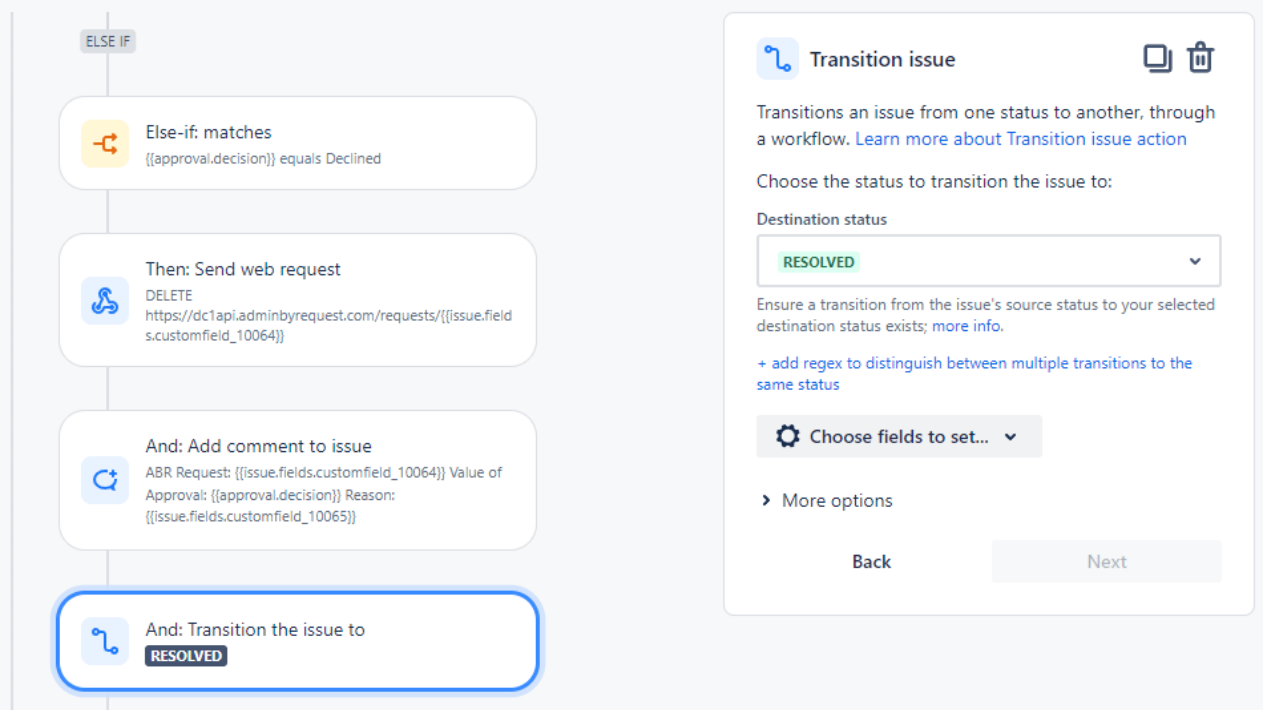
7. Action 1 - Send web request (DELETE). Important - make sure you read up on API calls here (API Overview) and here (Requests API):



8. Action 2 - Add Comment to Jira issue (Important - make sure you identify the correct customfield ids as described in "Overview" on page 1 - even if your field names are the same as the example, your customfield ids might be different):



9. Action 3 - Set Status to **Resolved**:



Document History

Version	Author	Changes
1.0 28 January 2024	Steve Dodson	Initial document release.
1.1 20 March 2024	Steve Dodson	Corrected typos and updated screenshots. Reorganized "Creating Jira Automation Rules" section.
2.0 17 April 2025	Steve Dodson	Updated manual structure and layout. Updated portal menu selections.

Index