



+



USE CASE:

Leveraging BetterCloud to fulfill and automate Self Service requests directly from Jira.

PRE REQS:

- Jira Administrator Permission Level
- BetterCloud Platform API
- Jira ServiceDesk Integration by BetterCloud installed in your BetterCloud tenant with an administrative level familiarity with Jira ServiceDesk
- Basic Understanding of building Custom Integrations in BetterCloud. Should you need a refresher, you can find more information in [this help center article](#).

SETUP:

BetterCloud

- Navigate to your Installed instance of Jira ServiceDesk in the BetterCloud Integration Center. From there, you can select “Modify” and navigate to “Extensions” to set up your API endpoint in BetterCloud for Jira to send its event data. Select “+ Add an Extension” to get started.
- From there, you will select, “Create a Trigger for Workflows” (fig 1)
- Once you have given the Trigger a name and optional description, you will be able to move onto creating the API endpoint.
- You can now copy the endpoint to your clipboard directly from the BetterCloud UI (fig 2)
- The next part of your setup will take place in Jira ServiceDesk

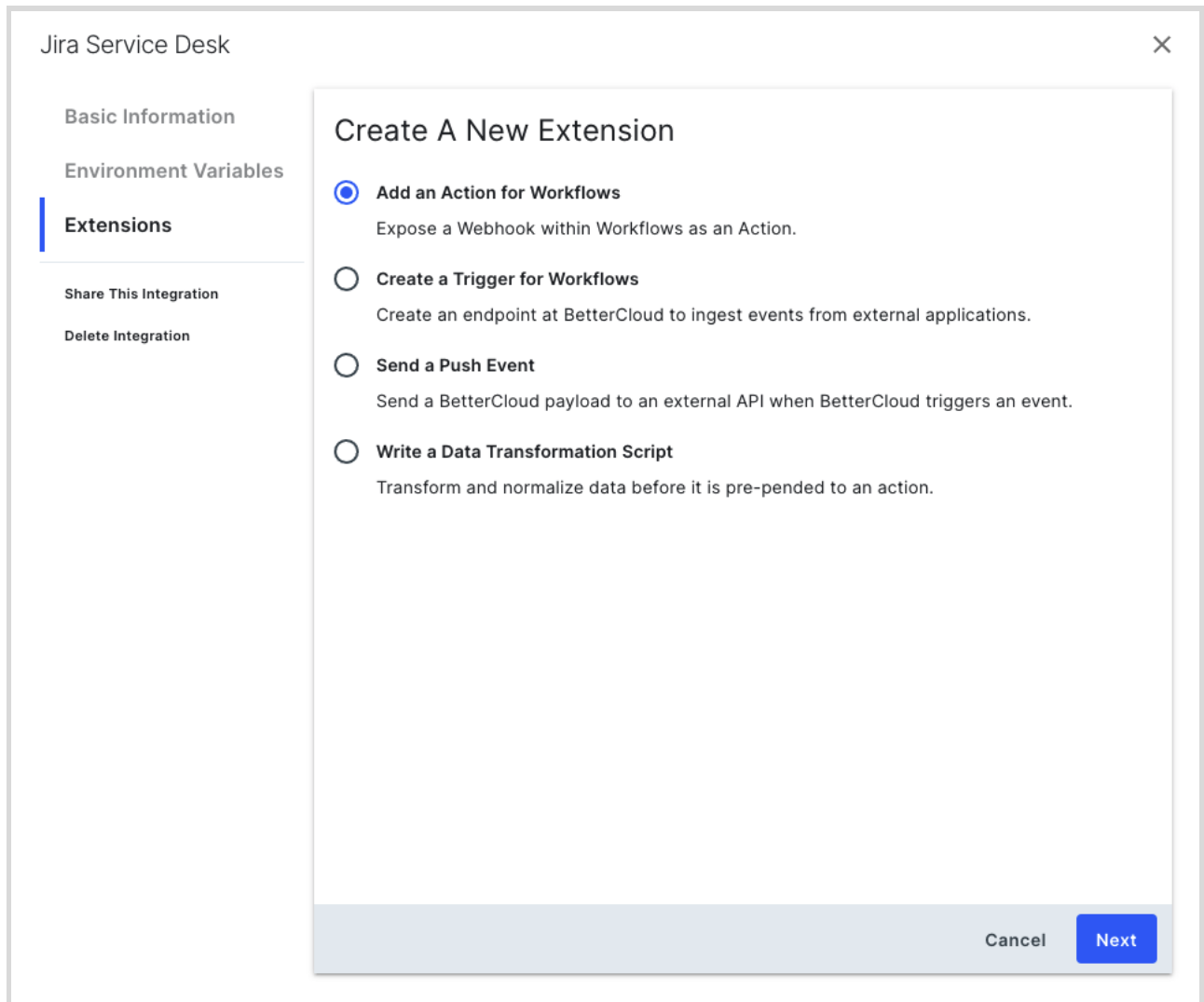


Fig 1

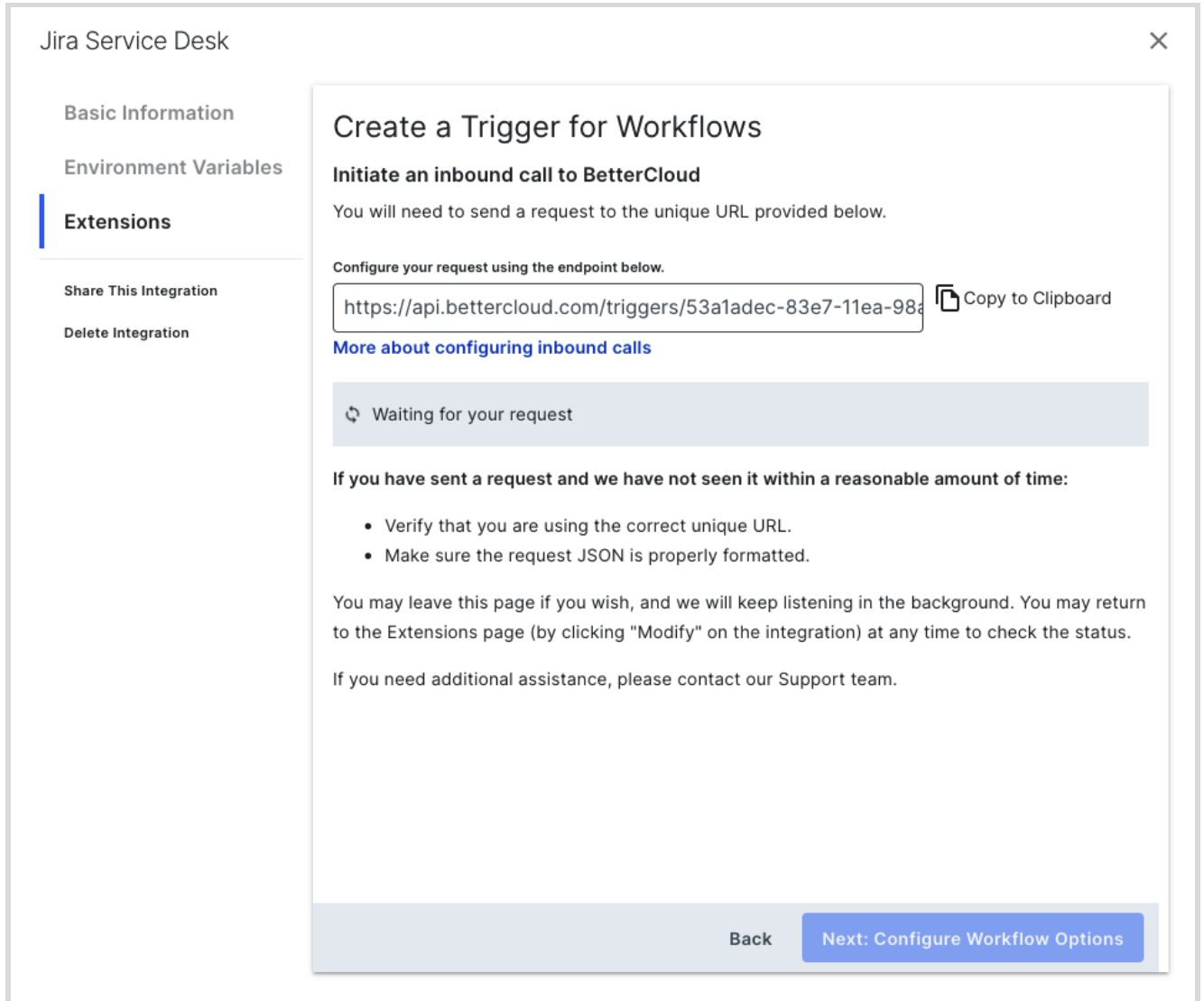


Fig 2

Jira ServiceDesk

- In Jira ServiceDesk, navigate to your “Project settings” and select “Automation”
- Select, “Create new automation”
- In your Jira automation, use the appropriate filters to target the specific request you will automate with BetterCloud
- From here, you will select “New Action” from the list of available components. You can filter the list of available components and search for, “Send web request” (fig 3)

- Once you add the component, you will paste the API endpoint you copied from BetterCloud while setting up the Custom Trigger. (fig 4)
- For the “Web request body”, you can either choose to send all of the “Issue data” or “Custom Data” which allows you to limit the amount of data sent to BetterCloud.
- You can add other customizations you need into your automation, or just save it as is. The rest of the setup will take place in BetterCloud
- You will need to call the API endpoint you have set up in BetterCloud in the previous section in order to complete the setup on the BetterCloud side. You can either validate the web request within that form by providing a relevant issue key, or manually make a service request that would trigger the automation.

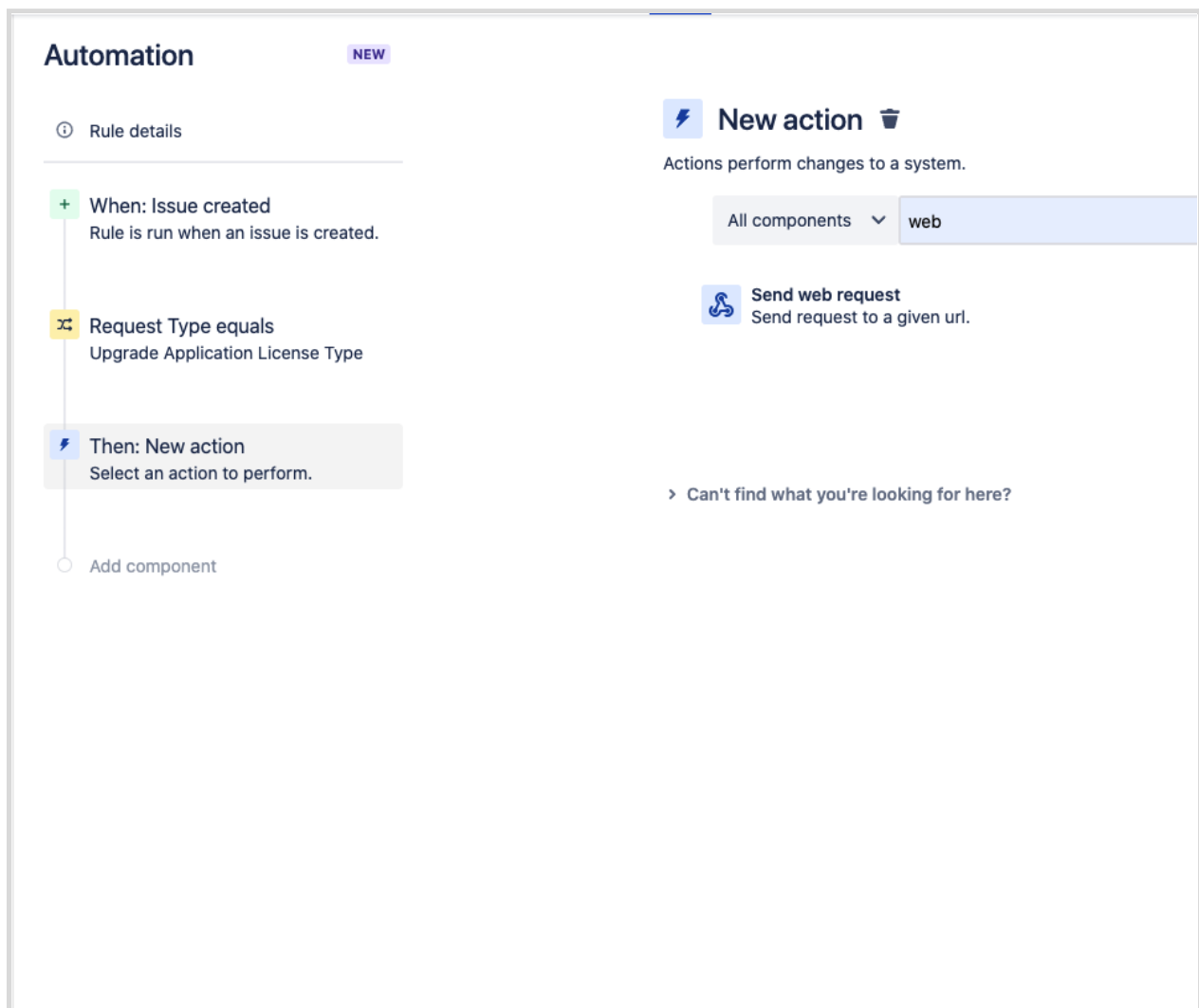


Fig 3

Automation NEW

ⓘ Rule details

+ When: Issue created
Rule is run when an issue is created.

⚙️ Request Type equals
Upgrade Application License Type

🔗 Then: Send web request
POST
https://api.bettercloud.com/triggers/53a1adec-83e7-11ea-98a2-2172b1d2ee47/6ad6f46d-28b0-11ec-9517-6bd2223186b7/613d76dd-0df3-4fb1-9368-92dcc524c92d?authorization=ONGr5wEkP4c7LCSksS

⊕ Add component

🔗 Send web request 🗑️

This action will send a HTTP request to the url specified below:

Web request URL *

https://api.bettercloud.com/triggers/53a1adec-83e7-11ea-98a2-2172b1d2ee47/6ad6f46d-28b0-11ec-9517-6bd2223186b7/613d76dd-0df3-4fb1-9368-92dcc524c92d?

Request parameters must be url encoded, smart values should use: {{value.urlEncode}}.

Headers (optional)

Name (e.g. Content-Type)	Value (e.g. application/json)
<p>➕ Add</p>	

HTTP method *

POST

Web request body *

Issue data

Wait for response

☐ Delay execution of subsequent rule actions until we've received a response for this web request

> Validate your web request configuration

Cancel Save

> How do I access web request response values in subsequent rule actions?

Fig 4

BetterCloud

- Return to the configuration page in BetterCloud for your Custom Trigger.
- If BetterCloud has not received an inbound trigger from Jira, you will see the page will state that it is, “Still waiting for your request”. If this is the case, you will need to ensure your automation is set up correctly so that it will send the webhook to BetterCloud when the conditions have been met. (fig 5)
- If you have successfully triggered the endpoint, you will be taken to a page that will allow you to determine which parameters that have been passed to BetterCloud should be available in the Workflow Builder. (fig 6)
- From here you can decide which parameters should be available as IF statements and which should be available as variables when performing actions within a BetterCloud workflow.
- You can now “Save & Publish” your new Custom Trigger.
- From here, you are ready to build a new workflow that will trigger from a Service Request being raised in Jira ServiceDesk.

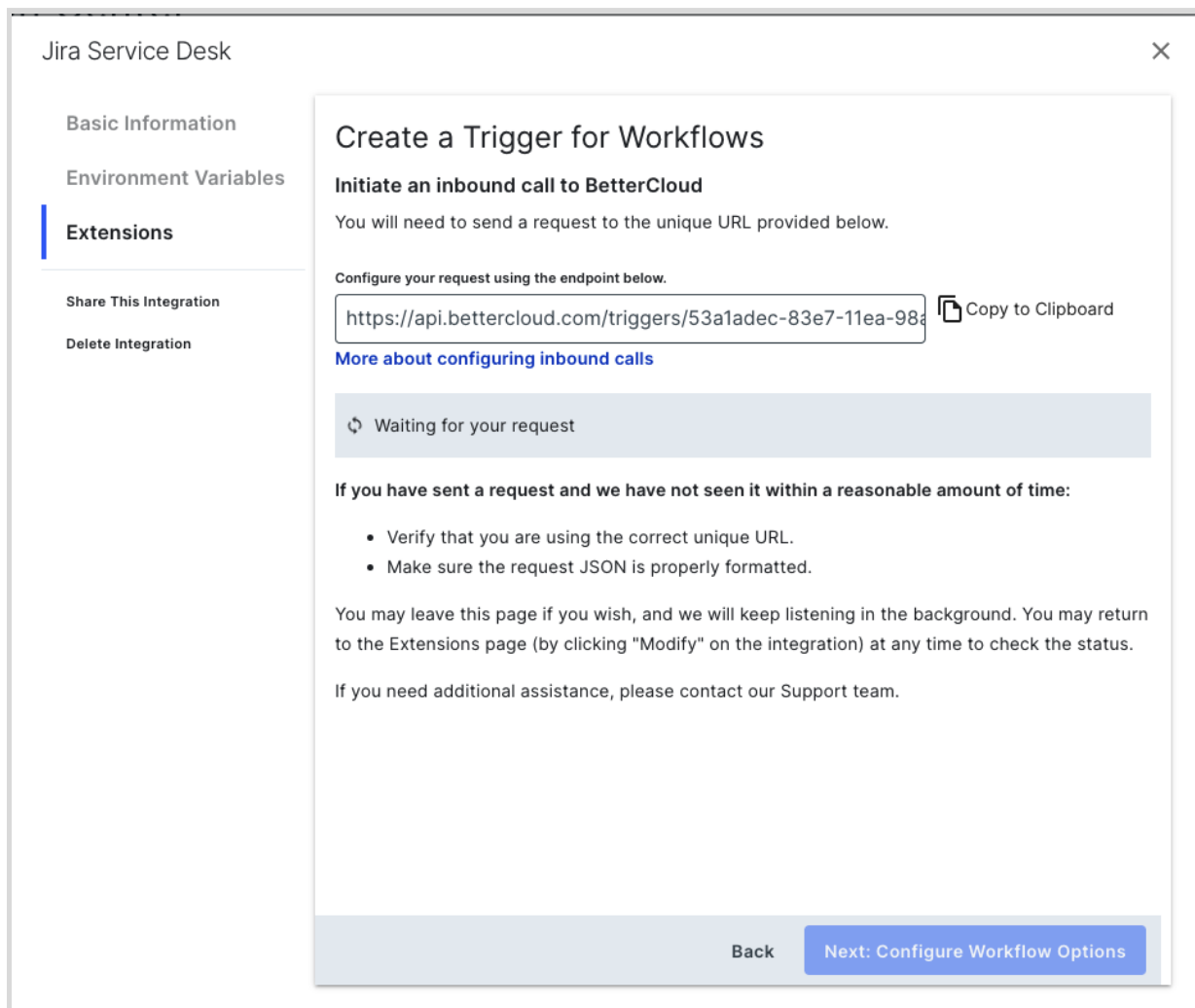


Fig 5

Jira Service Desk

Basic Information

Environment Variables

Extensions

Share This Integration

Delete Integration

Select which parameters you want to use for either Conditions or Dynamic Fields in your workflow.

Parameter	If Condition	Dynamic Field
self	<input type="checkbox"/>	<input type="checkbox"/>
id	<input type="checkbox"/>	<input checked="" type="checkbox"/>
key	<input type="checkbox"/>	<input checked="" type="checkbox"/>
changelog.startAt	<input type="checkbox"/>	<input type="checkbox"/>
changelog.maxResults	<input type="checkbox"/>	<input type="checkbox"/>
changelog.total	<input type="checkbox"/>	<input type="checkbox"/>
fields.customfield_10062.self	<input type="checkbox"/>	<input type="checkbox"/>
Application Name	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Back

Save & Publish

Fig 6

TIPS & TRICKS

Jira ServiceDesk was designed to give a great deal of flexibility around how requests are raised, approved, and managed. By combining the power and flexibility of incident management of Jira with the extensive API library in BetterCloud you can create some very sophisticated automations. Below, you will find a few examples:

- Leverage the built in Automation Scheduler in Jira ServiceDesk to “Schedule Offboarding” based on an End Date placed in a Jira request form.

- Setup API endpoints in JiraService desk with corresponding Custom Actions in BetterCloud to 'call back' to Jira ServiceDesk and perform actions on tickets such as closing out a request after BetterCloud has successfully completed the required Actions.
- Use the branching logic in Jira to enable requesters to assign different users for Files and Calendars to be transferred to during an offboarding
- Maintain existing approval processes in Jira ServiceDesk by only sending the Trigger to BetterCloud once any required approvals have been submitted.