

Jitbit Auto-Triage — Type Field + BookStack + Tech Note

Overview

Field	Value
Workflow ID	IIP1pezJvYAGKjYA
n8n URL	https://n8n.pbr.org.au/workflow/IIP1pezJvYAGKjYA
Status	Active
Trigger	Webhook — called by a Jitbit automation rule on every new ticket
Webhook URL	<code>https://n8n.pbr.org.au/webhook/jitbit-ticket-triage</code>
Systems	Jitbit Helpdesk (caller), Anthropic Claude API (classifier), BookStack (documentation search), Jitbit API (write-back)
AI Model	claude-haiku-4-5-20251001

Purpose

This workflow automatically triages every new Jitbit helpdesk ticket. When a ticket is created, Jitbit fires an automation rule that POSTs the ticket details to this n8n webhook. The workflow then:

1. Uses Claude Haiku (Anthropic API) to classify the ticket as an ITIL type and extract a triage summary and BookStack search keywords.
2. Sets the ticket's **Type** custom field in Jitbit to the classified value.
3. Searches BookStack for relevant internal documentation using the AI-extracted keywords.
4. Posts a private tech-only comment to the ticket containing the ITIL type, a triage summary, and links to relevant BookStack pages.

This gives attending technicians an immediate structured summary and relevant documentation links before they even open the ticket.

Jitbit Automation Rule

The workflow is triggered by a Jitbit automation rule (not a Jitbit AI External Tool — the trigger is a direct HTTP call, not routed through the Jitbit AI assistant). The rule is configured as:

Setting	Value
Trigger	Ticket is created
Action	Send HTTP request
Method	POST
URL	<code>https://n8n.pbr.org.au/webhook/jitbit-ticket-triage</code>
Post Data	<code>ticket_id=#ticketId#&subject=#subject#&body=#body#&category=#category#&tags=#tags#</code>

Jitbit substitutes the `#ticketId#`, `#subject#`, `#body#`, `#category#`, and `#tags#` tokens with the actual ticket values before sending the POST request.

How It Works

Step 1 — Receive webhook from Jitbit

The webhook node receives the POST body from Jitbit containing `ticket_id`, `subject`, `body`, `category`, and `tags`.

Step 2 — Classify with Claude

The workflow POSTs to the Anthropic Messages API (`https://api.anthropic.com/v1/messages`) using the `claude-haiku-4-5-20251001` model. The system prompt instructs Claude to act as an ITIL-aligned IT helpdesk triage assistant for PBR and respond with raw JSON only (no markdown). The user message contains the ticket subject, category, tags, and body.

Claude returns a JSON object with three fields:

Field	Description
<code>type</code>	ITIL classification — one of: Incident, Problem, Service Request, Change Request, Event
<code>bookstack_query</code>	3-6 keyword search terms for finding relevant internal documentation
<code>triage_summary</code>	2-3 sentence plain-English summary of the issue for the attending technician

The Parse Classification node strips any markdown code fences from the response (Claude Haiku occasionally wraps JSON in backticks despite instructions) before parsing the JSON.

Step 3 — Parallel branches

After parsing, two branches run in parallel:

- **Branch A — Set Type Custom Field:** POSTs to `https://helpdesk.pbr.org.au/api/SetCustomField` with `ticketId`, `fieldId=1` (the Type field), and the option ID corresponding to the classified type.
- **Branch B — Search BookStack:** Calls `https://bookstack.pbr.org.au/api/search` with the AI-extracted keywords, returning up to 5 results.

Both branches feed into a Merge node that waits for both to complete before continuing.

Step 4 — Build and post tech note

The Build Tech Note node constructs a private comment body combining the ITIL type, triage summary, and formatted BookStack links with content previews. The `ticket_id` and `triage` fields are read directly from the Parse Classification node output (not from the merge) to ensure they are never overwritten by the empty response body from the SetCustomField API call.

The comment is posted to Jitbit via `POST /api/comment` with `forTechsOnly=true`, making it visible only to technicians.

Step 5 — Respond

The workflow returns a JSON confirmation to Jitbit: `{ "result": "Triage complete. Type set to: X. Tech note posted to ticket." }`.

ITIL Type to Custom Field Option ID Mapping

The Jitbit Type custom field (Field ID: 1) uses dropdown option IDs. The mapping is hardcoded in the Parse Classification node:

ITIL Type	Option ID	Definition
Incident	1	Unplanned interruption to service (e.g. outage, crash, failure, offline, broken)
Problem	28	Underlying root cause investigation of one or more recurring incidents
Service Request	3	Routine pre-approved request (e.g. password reset, new laptop, software install)
Change Request	4	Planned alteration, addition, or removal of IT systems
Event	29	Automated monitoring alert (e.g. server monitoring trigger)

If Claude returns an unrecognised type value, the option ID defaults to 1 (Incident).

Example Tech Note Output

AI Triage

Type: Incident

Summary: The network printer at Belgrave station has been offline since 8am, preventing staff from printing boarding passes. Immediate investigation of printer connectivity and network status is required.

Relevant Documentation:

- Fault Finding - Belgrave Ticket Printers (page)

<https://bookstack.pbr.org.au/books/printers/page/fault-finding-belgrave-ticket-printers>

Issues with any of the Zebra ZD421 Belgrave Ticket printers where it is not possible to

ge...

- Printer Setup Guide (page)

<https://bookstack.pbr.org.au/books/endpoint-devices/page/printer-setup>

Steps to configure network printers at PBR sites...

Nodes

Node	Type	Purpose
Receive from Jitbit AI	Webhook (POST, responseMode: responseNode)	Receives ticket data from Jitbit automation rule
Classify with Claude	HTTP Request (POST)	Calls Anthropic API with ticket content; returns ITIL type, triage summary, and search query
Parse Classification	Set	Strips markdown fences; parses JSON; maps type string to option ID; extracts ticket_id and subject from webhook input
Set Type Custom Field	HTTP Request (POST)	Calls Jitbit <code>/api/SetCustomField</code> to set Field ID 1 to the classified type option ID
Search BookStack	HTTP Request (GET)	Searches BookStack with AI-extracted keywords; returns up to 5 results
Combine Triage + Docs	Merge (combineByPosition)	Waits for both parallel branches to complete
Build Tech Note	Set	Assembles the private comment body; reads triage fields from Parse Classification node directly to avoid merge overwrite
Post Tech Note to Jitbit	HTTP Request (POST)	Posts private tech-only comment to the ticket via Jitbit <code>/api/comment</code>
Respond to Jitbit	Respond to Webhook	Returns confirmation JSON to Jitbit

Credentials

Credential	Type	Used For
Anthropic API Key	HTTP Header Auth (<code>x-api-key</code>)	Claude Haiku API calls

Credential	Type	Used For
Jitbit API Token	HTTP Bearer Auth	SetCustomField and comment POST calls to Jitbit
BookStack Token	HTTP Header Auth (Authorization: Token id:secret)	BookStack search API

Known Quirks

- **Claude Haiku and markdown fences:** Despite the system prompt instructing raw JSON output, Claude Haiku 4.5 occasionally wraps its response in markdown code fences (`````). The Parse Classification node strips these defensively before parsing.
- **Merge node field overwrite:** The SetCustomField API returns an empty response body (`{}`), which causes a `combineByPosition` merge to overwrite the triage fields from the other branch. This is worked around by having the Build Tech Note node reference the Parse Classification node directly via `.item.json.*` rather than relying on `merge` output.
- **UpdateTicket does not support custom fields:** The Jitbit `/api/UpdateTicket` endpoint does not accept custom field parameters. The dedicated `/api/SetCustomField` endpoint must be used instead.

Maintenance Notes

- **AI model:** The workflow uses `claude-haiku-4-5-20251001`. Check Anthropic's model deprecation schedule periodically. To update the model, edit the `jsonBody` parameter in the Classify with Claude node.
- **Type option IDs:** If the Jitbit Type custom field options are changed (added, renamed, or reordered), update the option ID mapping object in the Parse Classification node. Current IDs can be verified via `GET https://helpdesk.pbr.org.au/api/customfields`.
- **Jitbit API token:** The token is stored as an HTTP Bearer Auth credential. Regenerate at `https://helpdesk.pbr.org.au/User/Token` if the token expires or is revoked.
- **Anthropic API key:** Stored as HTTP Header Auth with name `x-api-key`. Update in n8n credentials if the key is rotated.
- **BookStack credential:** Uses a `Token id:secret` format. Regenerate under the Claude_AI user in BookStack admin if access is revoked.

Revision #1

Created 2026-05-07 06:17:10 UTC by PBR_Documentation

Updated 2026-05-07 06:17:10 UTC by PBR_Documentation