

CX-Catalyst - Jira Integration Guide

Complete guide for configuring Jira as the bug tracking and escalation system for CX-Catalyst.

Table of Contents

1. [Overview](#)
 2. [Prerequisites](#)
 3. [Jira Project Setup](#)
 4. [Authentication & Credentials](#)
 5. [n8n Configuration](#)
 6. [Workflow Integration](#)
 7. [Field Mapping](#)
 8. [Escalation Workflow](#)
 9. [Troubleshooting](#)
-

Overview

CX-Catalyst integrates with Jira Cloud to:

- **Create bug tickets** automatically when recurring defects are detected (Workflow 3 & 5)
- **Escalate complex cases** that require engineering investigation
- **Track resolution** by linking Jira issues to support cases
- **Report trends** by correlating support volume with Jira backlogs

Jira integration is **optional** — the core support system functions without it, but escalation and bug tracking are significantly improved with Jira connected.

Prerequisites

- Jira Cloud account with admin access
 - Atlassian API token
 - A Jira project for support escalations
 - n8n instance with HTTP Request or Jira nodes available
-

Jira Project Setup

Create a Support Project

1. In Jira, go to **Projects > Create project**
2. Choose **Scrum** or **Kanban** (Kanban recommended for support)
3. Set **Project Key**: SUP (or your preferred key)
4. Set **Project Name**: "Support Escalations"

Configure Issue Types

Create or verify these issue types exist:

Issue Type	Purpose	Used By

Bug	Defects detected from support cases	Workflow 3 (Proactive Detection), Workflow 5 (Learning)
Task	General escalation actions	Workflow 4 (Collaborative Support)
Incident	Critical production issues	Workflow 1 (Intake, critical priority)

Add Custom Fields

Add a custom field to link Jira issues back to support cases:

1. Go to **Settings > Issues > Custom fields**
2. Click **Create custom field**
3. Select **Text Field (single line)**
4. Name: `Support Case ID`
5. Associate with the `SUP` project screens

Configure Workflow States

Recommended Jira workflow states for the SUP project:

```
Open → In Progress → In Review → Done
      ↓
Blocked → Reopened → In Progress
```

Authentication & Credentials

Get Your Cloud ID

Your Jira Cloud ID is required for API calls:

```
curl -u your-email@company.com:your-api-token \
https://your-domain.atlassian.net/_edge/tenant_info
```

The response includes your `cloudId`. Store this value — it's needed in n8n environment variables.

Create an API Token

1. Go to id.atlassian.com/manage-profile/security/api-tokens
2. Click **Create API token**
3. Name: `n8n CX-Catalyst`
4. Copy the token immediately (it won't be shown again)

Configure n8n Credentials

Option 1: Jira Node Credential

1. In n8n, go to **Settings > Credentials**
2. Click **Add Credential > Jira Software Cloud**
3. Enter:
 - o **Email:** Your Atlassian email
 - o **API Token:** The token from step above
 - o **Domain:** `your-domain` (without `.atlassian.net`)

4. Click **Save**

Option 2: HTTP Basic Auth (for REST API calls)

1. In n8n, go to **Settings > Credentials**
2. Click **Add Credential > HTTP Basic Auth**
3. Enter:
 - o **Name:** Jira HTTP Basic Auth
 - o **User:** Your Atlassian email
 - o **Password:** Your API token
4. Click **Save**

Set Environment Variables

In n8n Settings > Environment Variables:

```
JIRA_CLOUD_ID=your-cloud-id-here  
JIRA_PROJECT_KEY=SUP  
JIRA_BASE_URL=https://your-domain.atlassian.net
```

n8n Configuration

Using Jira Nodes

n8n provides dedicated Jira nodes:

- **Jira Software Cloud** — Create, update, and query issues
- **Jira Trigger** — Listen for Jira webhook events (issue created, updated, etc.)

Connect the credential created above to these nodes.

Using HTTP Request Nodes

For advanced operations not covered by the Jira node, use HTTP Request with the Jira REST API:

```
Base URL: https://your-domain.atlassian.net/rest/api/3  
Auth: HTTP Basic Auth (email + API token)  
Content-Type: application/json
```

Example: Create an issue via HTTP Request

```
{  
  "fields": {  
    "project": { "key": "{{ $env.JIRA_PROJECT_KEY }}" },  
    "summary": "Bug: {{ $json.issue_title }}",  
    "issuetype": { "name": "Bug" },  
    "description": {  
      "type": "doc",  
      "version": 1,  
      "content": [  
        {  
          "type": "paragraph",  
          "content": [  
            { "type": "text", "text": "{{ $json.description }}" }  
          ]  
        }  
      ]  
    }  
  }  
}
```

```

        ]
    }
]
},
"customfield_XXXXX": "{{ $json.case_id }}"
}
}

```

Replace `customfield_XXXXX` with your actual "Support Case ID" custom field ID.

Workflow Integration

Workflow 1: Smart Intake & Triage

- **When:** A case is classified as **Critical** priority
- **Action:** Creates a Jira Incident issue and links it to the support case
- **Notification:** Posts the Jira issue link in #support-alerts Slack channel

Workflow 3: Proactive Issue Detection

- **When:** Error spike or anomaly detected affecting multiple customers
- **Action:** Creates a Jira Bug with diagnostic details and affected customer count
- **Fields populated:** Summary, description, severity, affected customers, root cause hypothesis

Workflow 4: Collaborative Support Hub

- **When:** A case is rejected during human review (indicating a product defect)
- **Action:** Creates a Jira Bug or Task linked to the rejected case
- **Fields populated:** Summary, description, case history, reviewer comments

Workflow 5: Continuous Learning

- **When:** Daily analysis identifies recurring issues that suggest product bugs
- **Action:** Creates Jira Bug tickets with trend data and affected case count
- **Fields populated:** Summary, description, occurrence count, recommended fix

Field Mapping

Standard field mapping from CX-Catalyst to Jira:

CX-Catalyst Field	Jira Field	Notes
case_id	Support Case ID (custom)	UUID linking back to the case
Case description	Description	Formatted as Atlassian Document Format (ADF)
Case category	Labels	e.g., configuration, authentication
Case priority	Priority	Mapped: Critical→Highest, High→High, Medium→Medium, Low→Low

Customer tier	Labels	e.g., enterprise, smb
AI analysis	Description (appended)	Includes AI reasoning and suggested fix

Escalation Workflow

Automatic Escalation Flow

```

Support Case (Low Confidence or Rejected)
  |
  ▼
Create Jira Issue
  |
  ▼
Link Case ID to Jira Issue
  |
  ▼
Post Jira Link in #support-alerts
  |
  ▼
Update Case Status → "Escalated"
  |
  ▼
Engineering Resolves Jira Issue
  |
  ▼
(Manual) Update Support Case with Resolution

```

Jira Webhooks (Inbound)

Optionally configure Jira webhooks to notify CX-Catalyst when escalated issues are resolved:

1. In Jira, go to **Settings > System > WebHooks**
2. Add a webhook:
 - o **URL:** <https://your-n8n.com/webhook/jira/status-update>
 - o **Events:** Issue Updated (status changed to Done)
3. Build a Jira Trigger workflow in n8n to auto-close the linked support case

Troubleshooting

Issue	Cause	Solution
"Project not found"	Wrong project key	Verify project key in Jira (Settings > Projects)
"Issue type not found"	Mismatched issue type name	Check exact issue type names (case-sensitive)
401 Unauthorized	Invalid credentials	Regenerate API token; verify email matches

Field validation errors	Missing required fields	Use <code>getJiraIssueTypeMetaWithFields</code> to check requirements
Custom field not found	Wrong field ID	Look up field ID via Jira REST API: GET <code>/rest/api/3/field</code>
ADF formatting errors	Invalid description format	Use the Atlassian Document Format (ADF) builder or plain text fallback

Jira Integration Guide v1.0 - January 2026