

# n8n ? Zendesk Web Widget (Classic) – JWT Integration

Internal Documentation v1.0 – 2025-08-25

## “ Goal

Allow visitors authenticated through your n8n chat front-end to start a Zendesk Web Widget session, while still requiring a human agent to approve any ticket creation.

## 1. Prerequisites

Item	Where to find
Zendesk account with <b>Web Widget (Classic)</b> enabled	Admin Center → Channels → Widget
<b>Shared Secret</b> for JWT	Admin Center → Channels → Chat → Widget → <i>Authentication</i>
n8n instance reachable from the public internet	<code>https://your-n8n.com</code>
Existing n8n workflow that pauses for human review	(Human-in-the-Loop)

## 2. High-Level Flow

1. Visitor loads your web-chat.
2. Front-end requests a **JWT** from n8n.
3. n8n signs and returns the JWT.
4. Zendesk Web Widget starts an **authenticated chat** session.
5. **Human-in-the-Loop** still controls *ticket creation* (Wait node).

## 3. n8n Endpoints

### 3.1 JWT Issuer ( `POST /webhook/zendesk-jwt` )

**Purpose:** Zendesk will call this endpoint to verify the visitor.

## Workflow Steps

Node	Settings
Webhook	Path = <code>/webhook/zendesck-jwt</code> (POST)
Lookup User	Any node that confirms the <code>user_token</code> sent by Zendesk is valid (Database, Google Sheets, etc.).
JWT Sign	Algorithm = <code>HS256</code>
Respond to Webhook	Status = <code>200</code>

## Example cURL

```
curl -X POST \  
  'https://your-n8n.com/webhook/zendesck-jwt?user_token=abc123' \  
  -H 'Content-Type: application/json'
```

Expect:

```
{"jwt": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."}
```

## 3.2 Optional Token Generator for Front-End (GET `/webhook/chat-token`)

If your front-end needs to fetch the token itself (instead of letting Zendesk call the endpoint directly), create a second simple workflow:

Node	Settings
Webhook	Method = <code>GET</code>
JWT Sign	Same payload & secret as above
Respond to Webhook	Body = <code>{"jwt": "{{ \$('JWT Sign').item.jwt }}"}</code>

## 4. Zendesk Configuration

Admin Center → Channels → **Chat** → **Widget** → **Authentication**

Field	Value
Authentication Method	JWT
JWT URL	<code>https://your-n8n.com/webhook/zendesk-jwt</code>
JWT Secret	<i>Paste the same Shared Secret used in n8n</i>

## 5. Front-End Snippet

Add this after the Zendesk Web Widget script is loaded:

```
<script>
  // Replace with your n8n endpoint if you created /webhook/chat-token
  fetch('/api/n8n/get-chat-token', { credentials: 'include' })
    .then(r => r.json())
    .then(({ jwt }) => {
      zE('webWidget', 'chat:setJwtFn', callback => callback(jwt));
    });
</script>
```

“ If you let Zendesk call your endpoint directly, omit the fetch and simply use the JWT URL configured above.

## 6. Human-in-the-Loop Remains Intact

Your existing workflow already has a **Wait** node that pauses until an agent approves. Nothing in the JWT flow changes that—ticket creation only proceeds **after** the webhook resume call.

## 7. Security Checklist

- [ ] HTTPS only (n8n & Zendesk endpoints).
- [ ] Rotate Shared Secret periodically → update both Zendesk and n8n JWT node.
- [ ] Log every JWT issuance (`iat`, IP, `user_token`).
- [ ] Validate `user_token` strictly; deny unknown tokens immediately.

# 8. Troubleshooting Quick-Table

Symptom	Likely Cause	Fix
Widget shows "Unable to authenticate"	Wrong Shared Secret or expired <code>iat</code>	Check secret & timestamp
404 from Zendesk	Wrong JWT URL	Ensure <code>https://your-n8n.com/webhook/zendesk-jwt</code> is publicly reachable
"Invalid JWT format"	Payload missing required claims	Ensure <code>name</code> , <code>email</code> , <code>jti</code> , <code>iat</code> are present

# 9. Change Log

Date	Author	Notes
2025-08-25	DevOps	Initial draft based on Zendesk article <a href="#">4408838925082</a>

Revision #1

Created 25 August 2025 09:05:17 by Admin

Updated 25 August 2025 09:08:48 by Admin