

[Sign Up](#)[Sign In](#)[Search](#)**@n8n/chat** TS0.27.1 • Public • Published 5 months ago [Readme](#) [Code](#) Beta [6 Dependencies](#) [11 Dependents](#) [54 Versions](#)

n8n Chat

This is an embeddable Chat widget for n8n. It allows the execution of AI-Powered Workflows through a Chat window.

Windowed Example [n8n Chat Windowed](#)

Fullscreen Example [n8n Chat Fullscreen](#)

Prerequisites

Create a n8n workflow which you want to execute via chat. The workflow has to be triggered using a **Chat Trigger** node.

Open the **Chat Trigger** node and add your domain to the **Allowed Origins (CORS)** field. This makes sure that only requests from your domain are accepted.

[See example workflow](#)

Make sure the workflow is **Active**.

How it works

Each Chat request is sent to the n8n Webhook endpoint, which then sends back a response.

Each request is accompanied by an `action` query parameter, where `action` can be one of:

- `loadPreviousSession` - When the user opens the Chatbot again and the previous chat session should be loaded
- `sendMessage` - When the user sends a message

Installation

Open the **Webhook** node and replace `YOUR_PRODUCTION_WEBHOOK_URL` with your production URL. This is the URL that the Chat widget will use to send requests to.

a. CDN Embed

Add the following code to your HTML page.

```
<link href="https://cdn.jsdelivr.net/npm/@n8n/chat/dist/style.
<script type="module">
    import { createChat } from 'https://cdn.jsdelivr.net/r

    createChat({
        webhookUrl: 'YOUR_PRODUCTION_WEBHOOK_URL '
    });
</script>
```

b. Import Embed

Install and save n8n Chat as a production dependency.

```
npm install @n8n/chat
```

Import the CSS and use the `createChat` function to initialize your Chat window.

```
import '@n8n/chat/style.css';
import { createChat } from '@n8n/chat';

createChat({
  webhookUrl: 'YOUR_PRODUCTION_WEBHOOK_URL'
});
```

Vue.js

```
<script lang="ts" setup>
// App.vue
import { onMounted } from 'vue';
import '@n8n/chat/style.css';
import { createChat } from '@n8n/chat';

onMounted(() => {
  createChat({
    webhookUrl: 'YOUR_PRODUCTION_WEBHOOK_URL'
  });
});
</script>
<template>
  <div></div>
</template>
```

React

```
// App.tsx
import { useEffect } from 'react';
import '@n8n/chat/style.css';
import { createChat } from '@n8n/chat';

export const App = () => {
  useEffect(() => {
    createChat({
      webhookUrl: 'YOUR_PRODUCTION_WEBHOOK_U
    });
  });
};
```

```
    }, []);  
  
    return (<div></div>);  
};
```

Options

The default options are:

```
createChat({  
  webhookUrl: '',  
  webhookConfig: {  
    method: 'POST',  
    headers: {}  
  },  
  target: '#n8n-chat',  
  mode: 'window',  
  chatInputKey: 'chatInput',  
  chatSessionKey: 'sessionId',  
  metadata: {},  
  showWelcomeScreen: false,  
  defaultLanguage: 'en',  
  initialMessages: [  
    'Hi there! 🙌',  
    'My name is Nathan. How can I assist you today'  
  ],  
  i18n: {  
    en: {  
      title: 'Hi there! 🙌',  
      subtitle: 'Start a chat. We're here to',  
      footer: '',  
      getStarted: 'New Conversation',  
      inputPlaceholder: 'Type your question.'  
    },  
  },  
});
```

```
    },  
  });  
};
```

webhookUrl

- **Type:** string
- **Required:** true
- **Examples:**
 - `https://yourname.app.n8n.cloud/webhook/513107b3-6f3a-4a1e-af21-659f0ed14183`
 - `http://localhost:5678/webhook/513107b3-6f3a-4a1e-af21-659f0ed14183`
- **Description:** The URL of the n8n Webhook endpoint. Should be the production URL.

webhookConfig

- **Type:** { method: string, headers: Record<string, string> }
- **Default:** { method: 'POST', headers: {} }
- **Description:** The configuration for the Webhook request.

target

- **Type:** string
- **Default:** '#n8n-chat'
- **Description:** The CSS selector of the element where the Chat window should be embedded.

mode

- **Type:** 'window' | 'fullscreen'
- **Default:** 'window'
- **Description:** The render mode of the Chat window.
 - In `window` mode, the Chat window will be embedded in the target element as a chat toggle button and a fixed size chat window.
 - In `fullscreen` mode, the Chat will take up the entire width and height of its target container.

showWelcomeScreen

- **Type:** boolean
- **Default:** false
- **Description:** Whether to show the welcome screen when the Chat window is opened.

chatSessionKey

- **Type:** `string`
- **Default:** `'sessionId'`
- **Description:** The key to use for sending the chat history session ID for the AI Memory node.

chatInputKey

- **Type:** `string`
- **Default:** `'chatInput'`
- **Description:** The key to use for sending the chat input for the AI Agent node.

defaultLanguage

- **Type:** `string`
- **Default:** `'en'`
- **Description:** The default language of the Chat window. Currently only `en` is supported.

i18n

- **Type:** `{ [key: string]: Record<string, string> }`
- **Description:** The i18n configuration for the Chat window. Currently only `en` is supported.

initialMessages

- **Type:** `string[]`
- **Description:** The initial messages to be displayed in the Chat window.

Customization

The Chat window is entirely customizable using CSS variables.

```
:root {  
  --chat--color-primary: #e74266;  
  --chat--color-primary-shade-50: #db4061;  
  --chat--color-primary-shade-100: #cf3c5c;  
  --chat--color-secondary: #20b69e;  
  --chat--color-secondary-shade-50: #1ca08a;  
  --chat--color-white: #ffffff;
```

```
--chat--color-light: #f2f4f8;
--chat--color-light-shade-50: #e6e9f1;
--chat--color-light-shade-100: #c2c5cc;
--chat--color-medium: #d2d4d9;
--chat--color-dark: #101330;
--chat--color-disabled: #777980;
--chat--color-typing: #404040;

--chat--spacing: 1rem;
--chat--border-radius: 0.25rem;
--chat--transition-duration: 0.15s;

--chat--window--width: 400px;
--chat--window--height: 600px;

--chat--header-height: auto;
--chat--header--padding: var(--chat--spacing);
--chat--header--background: var(--chat--color-dark);
--chat--header--color: var(--chat--color-light);
--chat--header--border-top: none;
--chat--header--border-bottom: none;
--chat--header--border-bottom: none;
--chat--header--border-bottom: none;
--chat--heading--font-size: 2em;
--chat--header--color: var(--chat--color-light);
--chat--subtitle--font-size: inherit;
--chat--subtitle--line-height: 1.8;

--chat--textarea--height: 50px;

--chat--message--font-size: 1rem;
--chat--message--padding: var(--chat--spacing);
--chat--message--border-radius: var(--chat--border-rad
--chat--message--line-height: 1.8;
--chat--message--bot--background: var(--chat--color-wr
--chat--message--bot--color: var(--chat--color-dark);
```

```
--chat--message--bot--border: none;
--chat--message--user--background: var(--chat--color-s
--chat--message--user--color: var(--chat--color-white)
--chat--message--user--border: none;
--chat--message--pre--background: rgba(0, 0, 0, 0.05);

--chat--toggle--background: var(--chat--color-primary)
--chat--toggle--hover--background: var(--chat--color-p
--chat--toggle--active--background: var(--chat--color-
--chat--toggle--color: var(--chat--color-white);
--chat--toggle--size: 64px;
}
```

Caveats

Fullscreen mode

In fullscreen mode, the Chat window will take up the entire width and height of its target container. Make sure that the container has a set width and height.

```
html,
body,
#n8n-chat {
  width: 100%;
  height: 100%;
}
```

License


You can find the license information [here](#)

Keywords

none

Provenance

Built and signed on

 **GitHub Actions**

[View build summary](#)

Source Commit

github.com/n8n-io/n8n@3c75565

Build File

<.github/workflows/release-publish.yml>


Public Ledger

[Transparency log entry](#)

[Share feedback](#)

Install

```
> npm i @n8n/chat
```



Repository

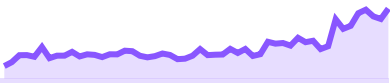
 github.com/n8n-io/n8n

Homepage

 n8n.io

Weekly Downloads

1,710



Version

0.27.1 

License

SEE LICENSE IN LICENSE.md

Unpacked Size

1.07 MB

Total Files

55

Issues

[450](#)

Pull Requests

[274](#)

Last publish

18 days ago

Collaborators



> Try on RunKit

🚩 Report malware



Support

Help

Advisories

Status

Contact npm

Company

About

Blog

Press

Terms & Policies

Policies

Terms of Use

Code of Conduct

Privacy