

[Menu](#)[On this page](#)

In addition to being used as a standalone application, Opsloom can also be embedded into existing websites as a widget. This tutorial will guide you through the embedding process so you can take full advantage of Opsloom's capabilities.

## Prerequisites

1. You will need an industry-specific hosted API endpoint that has been set up for you through the Opsloom team. If you don't have one, you can use the dummy endpoint provided below.
2. You will need access to modify the [index.html](#) of your host site. Opsloom works on any client-side framework!

## How to Embed Opsloom Chat

In any editor of your choice, open up the [index.html](#) file of the host site you manage.

Within [index.html](#), insert the following three lines. The `script` and `link` tags should go somewhere in your head, while the `div` can go in the body, under whatever component you like.

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <link rel="icon" type="image/svg+xml" href="/favicon.svg" />
```

html

```
<meta name="viewport" content="width=device-width, initial-scale=1.0"
<title>My Organization Website</title>

<script type="module" crossorigin src="https://opsloom.com/opsloom/op
<link rel="stylesheet" crossorigin href="https://opsloom.com/opsloom/
</head>
<body>
  <div id="root"></div>
  <script type="module" src="/src/main.jsx"></script>
  <div id="opsloom-chat" data-api-url="https://opsloom.com/opsloom-api/
</body>
</html>
```

### TIP

Although we provide an `index.html` file here, you can embed Opsloom in any file on your client-side app— `App.tsx` if you are using React, `layout.tsx` if you are using Next.js, etc. Just make sure to use the `id="opsloom-chat"` convention described above and use the `fullscreen` param in the root div to take up a whole container (for example, to occupy an entire sidebar).

### WARNING

You will need to replace the `data-api-url` with your own custom endpoint created in collaboration with the Opsloom team.

---

## Customization

There are several properties you can add to `<div id="opsloom-chat"/>` to suit your needs. All of the below are optional.

Property	Value	Description
<code>data-api-url</code>	<code>string</code>	This is the engine behind your Opsloom chatbot. You can set this up with someone on the Opsloom team for maximum flexibility. Alternatively, you can

Property	Value	Description
		use <a href="https://opsloom.com/">https://opsloom.com/</a> . If not specified, will assume the API is self-hosted on the same domain as the embedding, aka at <a href="https://yoursite.com/opsloom-api/v1">https://yoursite.com/opsloom-api/v1</a> .
<code>data-module-type</code>	<code>string</code>	<code>webapp</code> : Add this to make your whole app behave like an Opsloom chat. Note that this may have some adverse effects if other content is not coordinated correctly. For instance, this uses <code>vh</code> and <code>vw</code> rather than a CSS container, so it will attempt to take up the entire viewport. <code>fill-parent</code> : This treats the parent component as the container for the chatbot. Use this to create a custom component on your own website (for example, a drawer, modal, etc.). <code>bubble-widget</code> : This creates a small assistant in the bottom right corner that is fixed on every page of your app.
<code>data-forced-theme</code>	<code>"light"</code> or <code>"dark"</code>	If specified, it will force the component to use a particular theme. Otherwise the component will use the browser's default theme.
<code>data-headless</code>	<code>none</code>	Removes the history, control panel, etc.
<code>data-landing-path</code>	<code>string</code>	On load, the application will jump to the specified page by default. Useful with single-assistant apps.

For example, your div may look like:

html

```
<div id="opsloom-chat" data-module-type="bubble-widget" data-api-url="htt
```

WARNING

Your `id` MUST be equal to `"opsloom-chat"` or the DOM will have no way to connect to the chatbot.

Next Steps

That's it! You're good to go. Make sure your custom API is properly working and there are no CORS issues. If so, you'll need to be added to the "allow" list by reaching out to our team.

---

Previous page

[Styling](#)

Next page

[RAG AI Assistant](#)