

Add a Voice Agent Widget to Your Website

Spokiva Widget Integration Guide

Add your Spokiva voice agent to any website with a single script tag. Visitors click a button and talk to your AI agent right in the page — no phone number, no telephony setup required. This guide is written for the developer who will add the widget to a client website.

How it works

- The widget is a small JavaScript file that loads from our servers and renders a button (or an inline panel) on your page.
- When a visitor clicks the button, the widget connects to the voice agent and streams audio directly in the browser using WebRTC.
- No phone number is needed. Everything happens inside the visitor's browser.
- All configuration (button text, color, position, allowed domains) is stored on our side. Changes you make in the Spokiva admin panel apply automatically without re-embedding the script.

Where to find your embed code

1. Log in to the Spokiva admin panel: <https://admin.spokiva.neptuneinfotech.com>
2. Open your workflow (the voice agent you built).
3. Open **Settings** → **Configure Widget**.
4. Turn on **Enable Embedding**.
5. Optionally enter the **Allowed Domains** where the widget may run (leave empty to allow any website).
6. Click **Save Configurations**.
7. Copy the **Embed Code** shown below the button.

The embed script

The embed code looks like this:

```
<!-- Spokiva Widget -->
<script>
  (function(d, s, id) {
    var js, fjs = d.getElementsByTagName(s)[0];
    if (d.getElementById(id)) return;
    js = d.createElement(s); js.id = id;
    js.src = 'https://call.neptuneinfotech.com/embed/spokiva-widget.js?
token=emb_YOUR_TOKEN&environment=production&apiEndpoint=https://call.neptuneinfotech.com';
    js.setAttribute('data-spokiva-context', JSON.stringify({
      page_url: window.location.href,
```

```
    today: new Date().toISOString().slice(0, 10)
  }));
  js.async = true;
  fjs.parentNode.insertBefore(js, fjs);
})(document, 'script', 'spokiva-widget');
</script>
```

Copy the whole block and paste it just before the closing `</body>` tag of your website's HTML, or anywhere in the page you want the widget available.

*Note: your script will contain your own unique `token` . Do not share it — anyone with the token can embed your agent. Use the **Allowed Domains** list to restrict where it can run.*

Widget types

You can choose what kind of widget to show:

- **Voice Agent:** Visitors click a button and talk to the agent by voice
- **Chat Agent:** Visitors type messages to the agent

Embed modes

- **Floating Widget:** A button floats in a corner of the page. Most common.
- **Inline Component:** The widget renders inside a container you place in your page content.
- **Headless (Bring Your Own UI):** No built-in UI. Drive the agent from your own buttons via the JavaScript API.

Floating Widget

Shows a round button in a corner. Click it to open the voice or chat panel.

Configuration you can set in the admin panel:

- **Position:** bottom-right, bottom-left, top-right, or top-left.
- **Button Text:** the label on the button (e.g. "Talk to Agent").
- **Button Color:** the button's background color.

Inline Component

The widget renders inside a `<div>` you place in your page. Use this when you want the agent embedded in your page content rather than floating.

Add this where you want the widget to appear:

```
<div id="spokiva-inline-container" style="min-height: 480px">
  <!-- Widget renders here; no extra JS needed -->
```

</div>

For a voice agent, call these from your own buttons:

- `window.SpokivaWidget.start()` — begin the call
- `window.SpokivaWidget.end()` — end the call

Headless mode

No built-in UI. You build the interface and call the agent's JavaScript API.

This is for developers who want full control over the look and feel.

Voice agent API:

```
// start the call (must run inside a user click so the browser allows the mic)
window.SpokivaWidget.start();
// end the call
window.SpokivaWidget.end();
// status: idle, connecting, connected, failed
window.SpokivaWidget.onStatusChange((status) => { /* ... */ });
window.SpokivaWidget.onCallStart(() => { /* ... */ });
window.SpokivaWidget.onCallEnd(() => { /* ... */ });
window.SpokivaWidget.onError((err) => { /* ... */ });
```

Chat agent API:

```
window.SpokivaWidget.startChat(); // start the conversation
const transcript = await window.SpokivaWidget.sendMessage('Hello'); // send a message
window.SpokivaWidget.getMessages(); // read the transcript
window.SpokivaWidget.onMessage((text, turn) => { /* ... */ });
window.SpokivaWidget.onChatStateChange((state) => { /* ... */ });
// states: idle, starting, ready, waiting, ended, expired, error
```

Example — track call status with your own UI:

```
let callStatus = 'idle';

window.SpokivaWidget?.onStatusChange((status) => {
  callStatus = status;
  // re-render your UI here
});

document.getElementById('talk-btn').addEventListener('click', () => {
  if (callStatus === 'connected' || callStatus === 'connecting') {
    window.SpokivaWidget.end();
  }
});
```

```
} else {  
  window.SpokivaWidget.start();  
}  
});
```

Passing visitor details to the agent

You can pass information about the current visitor to the agent. This data is available inside your agent's prompts as `{{initial_context.<name>}}`.

The embed script already includes a `data-spokiva-context` attribute with `page_url` and `today`. Edit it or add your own fields:

```
js.setAttribute('data-spokiva-context', JSON.stringify({  
  page_url: window.location.href,  
  today: new Date().toISOString().slice(0, 10),  
  user_email: 'john@example.com',    // your own fields  
  plan: 'premium'  
}));
```

For details your page learns after load (for example a user's email typed into a form), call `setContext` before starting the conversation:

```
window.SpokivaWidget.setContext({  
  user_email: 'john@example.com',  
  plan: 'premium'  
});
```

Then reference them in your agent prompts as `{{initial_context.user_email}}`.

Allowed domains

The **Allowed Domains** list controls which websites may load the widget. If you leave it empty, the widget works on any website. To restrict it, add domains one at a time, for example `example.com` or `*.example.com`.

This is a security control: anyone in possession of your embed token can use it only on the domains you allow.

Browser and mic permission

The voice widget needs microphone access, which the browser only grants after a user action (a click). This is why the call starts when the visitor clicks the button — it is a browser requirement, not a limitation of the widget.

- The site must be served over **HTTPS** for microphone access to work in modern browsers (secure context).
- Desktop and mobile browsers are supported.

Changing configuration after embedding

You do **not** need to re-embed the script to change button text, color, position, widget type, or embed mode. Save the new configuration in the admin panel and it applies automatically — the widget fetches its configuration at load time.

Troubleshooting

- **Widget does not appear:** Confirm **Enable Embedding** is on in the admin panel. Check the browser console for a network error loading the widget script. Confirm the site is HTTPS.
- **"Failed to fetch config" in the console:** Your token may be inactive, expired, or the domain is not in the **Allowed Domains** list. Re-save the configuration in the admin panel.
- **Microphone prompt does not appear:** The call must start from a user click. Make sure `start()` is called inside a click handler and the page is HTTPS.
- **Audio is not heard:** Confirm the visitor's browser allows the microphone. Test in a private/incognito window.

Support

For help, contact your Spokiva representative or reach out through the Neptune Infotech team. Include the workflow name and a description of the issue.