

Chatbase Agent Setup and Deployment Guide

Ireru Linus Mugendi

July 7, 2025

Overview

This document outlines the comprehensive steps taken to build, train, and deploy a chatbot agent using **Chatbase**, including embedding it as a widget on a website or integrating it into a Next.js app.

1. Sign In and Workspace Setup

Step 1: Visit <https://www.chatbase.co/> and click **Sign In** using your preferred authentication method (e.g., Google, GitHub, email).

Step 2: Once authenticated, you'll be redirected to your Chatbase dashboard where you can manage agents.

2. Create a New Agent

Step 3: On the dashboard, click **Create a new bot**.

Step 4: Name your agent, set the welcome message, and define its purpose.

Step 5: Optionally configure the chatbot personality in the prompt section.

3. Train Agent with Knowledge

Step 6: Navigate to the **Sources** tab.

Step 7: Choose from multiple options:

- **Upload Files:** Drag and drop or browse PDFs, DOCX, or TXT files.
- **Paste Text Snippets:** Add short FAQs, glossary entries, or instructions.
- **Website or Sitemap Crawling:** Provide a URL or sitemap for auto-parsing.
- **Notion Integration:** Sync structured content from Notion.

Step 8: Once data is uploaded or linked, Chatbase will automatically index and retrain the agent.

4. Get the API Key

Step 9: Head over to the **API** tab in the Chatbase dashboard.

Step 10: Click **Generate Key**.

Step 11: Copy and store this API key securely — it will be used for authenticated requests.

5. Generate Embed Code

Step 12: Under the **Embed on your site** or **Website Widget** section, click **Generate Code**.

Step 13: You will receive a JavaScript snippet that includes:

- Your agent's unique ID
- Styling and placement controls

Step 14: Copy the entire script tag.

6. Integrate Embed Code into Website

Step 15: For static HTML sites: Paste the script before the closing `</body>` tag in your main HTML file.

Step 16: For Next.js apps: Insert it in the `layout.tsx` or `_app.tsx` file — anywhere that persists across all pages. A good place is inside a `useEffect` hook using `dangerouslySetInnerHTML` in React.

Step 17: Make sure the code runs client-side only to avoid SSR issues.

7. Optional Enhancements (May Be Forgotten)

- Configure **Chat Widget Theme** (colors, avatar, branding).
- Set response limits, fallback behavior, and memory options.
- Connect webhooks or Zapier for deeper integrations.
- Monitor traffic, retention, and chat logs via **Analytics**.

8. Benefits of Integrating a Chatbox into a Website

Integrating a chatbox into your website or application offers significant operational and customer experience advantages. The following points summarize its key benefits:

- **24/7 Customer Support:** Always-on assistance improves customer trust and accessibility.
- **Instant Query Resolution:** Reduces response time and enhances user satisfaction.
- **Higher Conversion Rates:** Guides users through product pages and the checkout process to boost sales.
- **Reduced Operational Costs:** Automates repetitive queries, minimizing human support effort.
- **Scalable Support:** Handles thousands of simultaneous interactions without increasing overhead.
- **AI-Driven Personalization:** Learns from user behavior to tailor recommendations and interactions.
- **Data Collection:** Captures insights on common pain points, user behavior, and engagement metrics.
- **Multilingual Communication:** Serves diverse audiences with language detection and translation.
- **Streamlined Navigation:** Acts as a virtual guide to help users access content or features quickly.
- **Cross-Platform Consistency:** Maintains consistent support across web, mobile, and social platforms.
- **Automated Lead Qualification:** Screens visitors with contextual questions to identify potential customers.

- **Transaction Automation:** Facilitates bookings, purchases, and form submissions directly in-chat.
- **Improved Accuracy:** Eliminates human error in repetitive or rules-based support scenarios.
- **Brand Engagement:** Provides a consistent voice, personality, and tone aligned with brand identity.
- **Continuous Improvement:** AI chatbots get smarter over time via self-learning and feedback loops.

Conclusion

By following the above steps, you've successfully launched a scalable, trained AI chatbot and embedded it directly into your website. You can continue evolving it by feeding more content or refining its prompt style for tone and UX.