spirebot

USER MANUAL

Build, publish, and run your bots.

A step-by-step guide to the Spire Bot workspace, with screenshots of every main screen.

November 2026 edition

Contents

1 Introduction

2 Getting started

3 Working with bots

4 Designing flows

5 Content and data

6 Bot settings

7 Validating and testing

8 Publishing and deploying

9 Live support

10 Operating and improving

11 Security and governance

12 Troubleshooting

13 Launch checklist

About the screenshots. The screenshots show a sample workspace. Names, email addresses, and data on your screens will differ, and you only see the menu items your role allows.

CHAPTER 1

Introduction

Spire Bot is a workspace for designing, publishing, and operating conversational bots for web chat and WhatsApp, with a live inbox for the people who take over when a customer needs a human.

Who this manual is for

- **Bot Designers** who design flows, test the bot, and publish it to channels.
- **API Integrators** who set up connections and the APIs a bot calls.
- **Agents and supervisors** who answer customers in the Agent inbox.
- **Security Admins** who create users and assign their roles.
- **Administrators** who manage departments, service health, and everything else.

How a bot is built

Treat a bot as a versioned product. Every release moves through the same six stages.

Stage	What you do
1. Plan	Define the audience, the tasks, the boundaries, the success signals, and when to hand over to a person.
2. Design	Turn each customer goal into a small, connected flow with a clear outcome.
3. Configure	Declare variables, content, languages, knowledge, scripts, and approved connections.
4. Test	Validate the structure, run conversations in the playground, and exercise failure paths.
5. Publish	Create a versioned release, deploy it to a channel, and test the live path.
6. Improve	Review conversations, analytics, and untrained queries, then fix gaps in the draft and release again.

> NOTE
>
> Your edits are saved to a **draft**. Customers keep talking to the last **published** version until you publish again.

Key terms

Term	Meaning
Workspace	Your organisation’s area. It holds the bots, users, departments, and API connections.
Bot	One assistant with its own flows, content, settings, deployments, and history.
Flow	A canvas of connected steps that handles one customer goal.
Step	One action in a flow, such as sending text, asking a question, or calling an API.
Draft	The editable copy of a bot. Changes stay here until you publish.
Release	A numbered, unchangeable published version of a bot.
Deployment	A channel, such as web chat or a WhatsApp number, that serves the published version.
Handover	Passing a conversation from the bot to a department of live agents.

CHAPTER 2

Getting started

Sign in, find your way around, and set up your own profile.

Sign in

1. Open your Spire Bot address in a browser.
2. Enter your email address and password.
3. Select **Sign in**.

Accounts are created by an Administrator or Security Admin. If you cannot sign in, ask them to check your account.

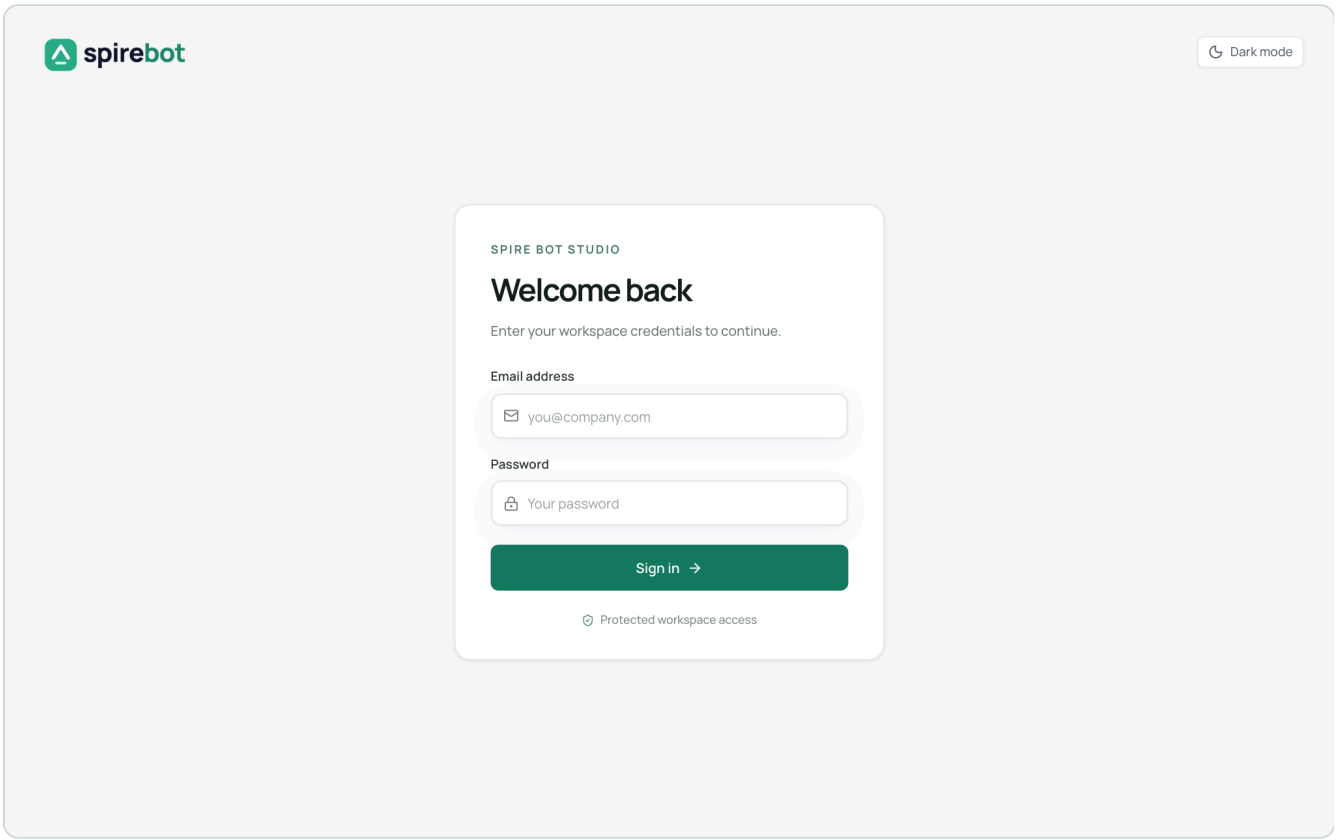


Figure 1. The sign-in page.

The dashboard

The dashboard opens after you sign in if your roles include the Studio. Otherwise you land on **Users & agents** (Security Admin) or the **Agent inbox**. Shortcut cards take you to the most common tasks: **Create** a bot, **Design** in the flow studio, open the **Support** inbox, review **Insights**, and manage the **Team**. Below them you see how many chatbots exist, how many are published, and your most recent bots.

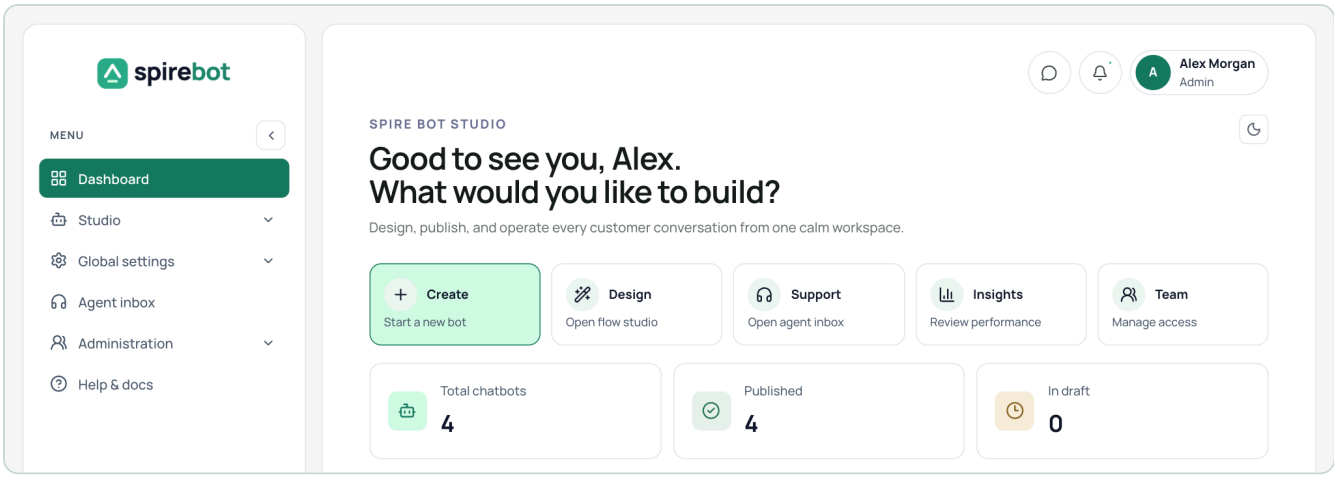


Figure 2. The dashboard with the main menu on the left.

The main menu

Menu item	What it opens
Dashboard	The overview page.
Studio	**All bots**, and for the bot you have open: Design, APIs, Translations, Scripts, Knowledge, Untrained queries, Deploy, Conversations, Analytics, Releases, Bot settings, and Audit log.
Global settings	**API connections** shared by every bot in the workspace.
Agent inbox	Live conversations handed over to your departments.
Administration	Users & agents, Departments, Chat tags, Canned responses, and Observability.
Help & docs	The online guide.

> **NOTE**
> The menu shows only what your roles allow. For example, Global settings appears for API Integrators and Administrators, and Administration appears for Security Admins and Administrators. See **Roles** in the Live support chapter.

- Use the arrow at the top of the menu to collapse it to icons and gain working space.
- Your name at the bottom of the menu opens your profile. The icon beside it signs you out.
- The moon button switches between light and dark mode. In the studio, the **A** button changes the text size.

You only see the menu items your role allows.

Your profile

Open your profile from your name at the bottom of the menu. Customers never see your account name or email. They see the nickname and avatar you set here when you reply in live chat.

1. Select **Upload avatar** and choose a PNG, JPG, or WebP image. It is cropped to a square.
2. Enter a **Nickname** of up to 40 characters. Leave it blank to show your account name.
3. Check the **Customer preview**, then select **Save profile**.

The Account panel shows your name, email, workspace roles, concurrent chat limit, and departments. These are managed by your workspace administrator.

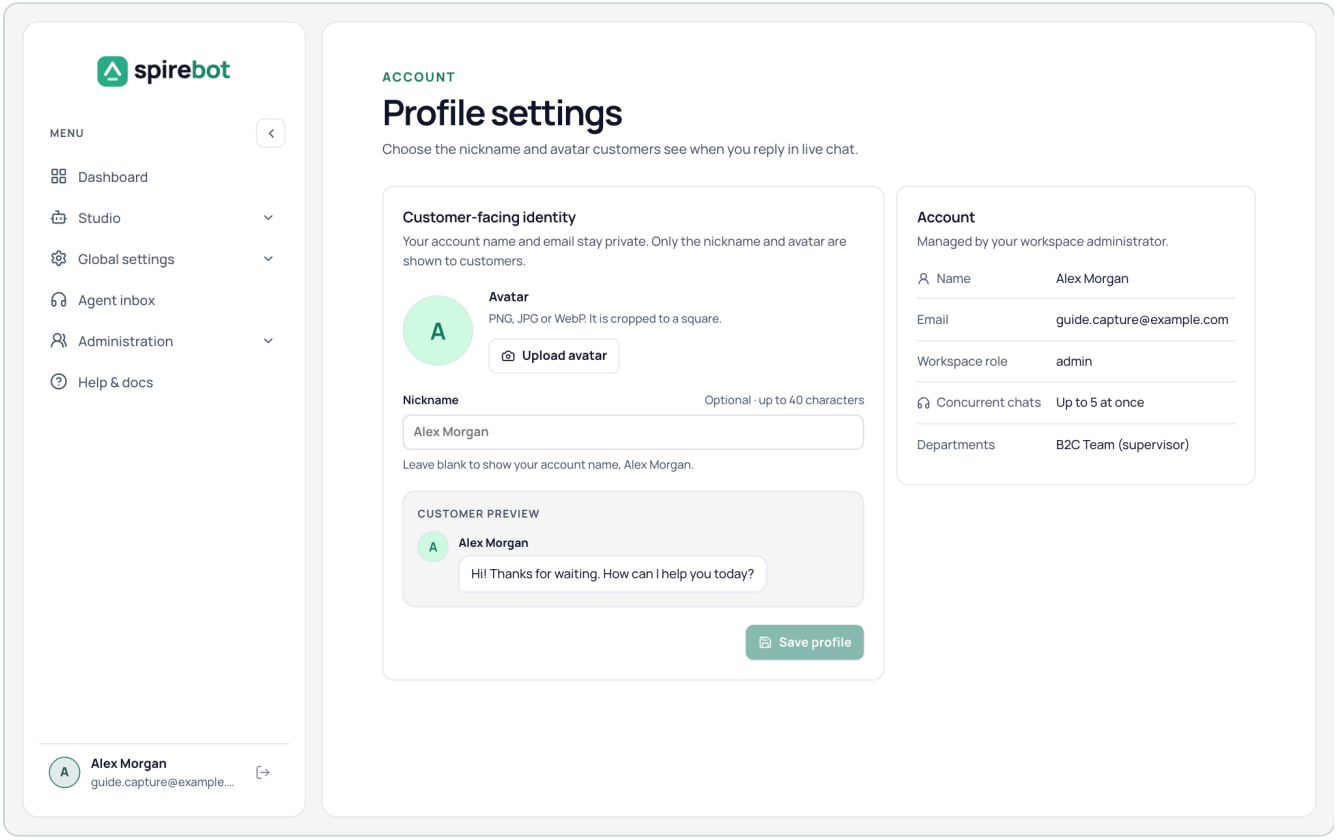


Figure 3. Profile settings with the customer preview.

CHAPTER 3

Working with bots

Every bot in the workspace is listed under Studio → All bots.

Each card shows the bot's name, its published version, its ID, and when it was created. Search by name, or change the sort order with the **Created** control. Select a card to open the bot in the designer.

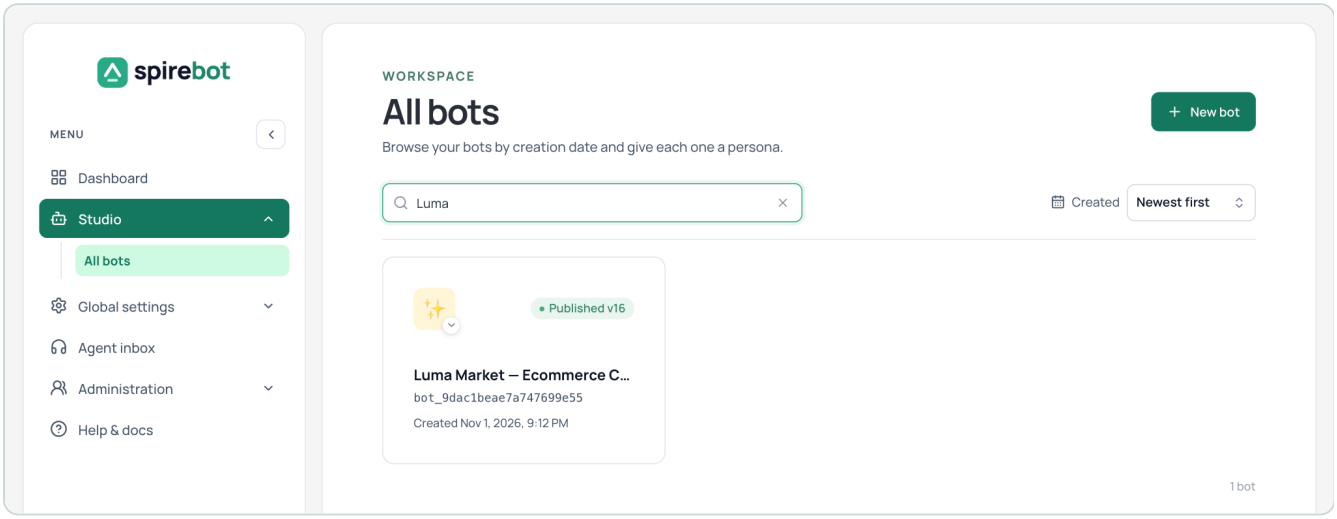


Figure 4. All bots, filtered with the search box.

Create a bot

1. Select **New bot** on the All bots page, or **Create** on the dashboard.
2. Give the bot a name that describes its job.
3. Open the bot and start with its first flow.

Move a bot between workspaces

Use **Export bot** and **Import bot** in the designer to copy a bot from one workspace to another.

Plan before you build

A strong bot is explicit about who it serves, what it can do, and when it should stop or transfer the conversation. Write a short brief first.

- **Audience:** who is speaking, in which channels and languages?
- **Top tasks:** the three to seven outcomes that matter most.
- **Boundaries:** requests the bot must refuse, redirect, or hand off.
- **Success:** how you will measure completion, handover, and failure.
- **Ownership:** who owns the content, who publishes, and which team answers live chats.

For every customer goal, sketch the shortest happy path, the likely wrong inputs, the path when a service is unavailable, and the route to a person. Keep one responsibility per flow.

CHAPTER 4

Designing flows

The flow designer is where you build the conversation: a canvas of steps joined by connections.

The designer at a glance

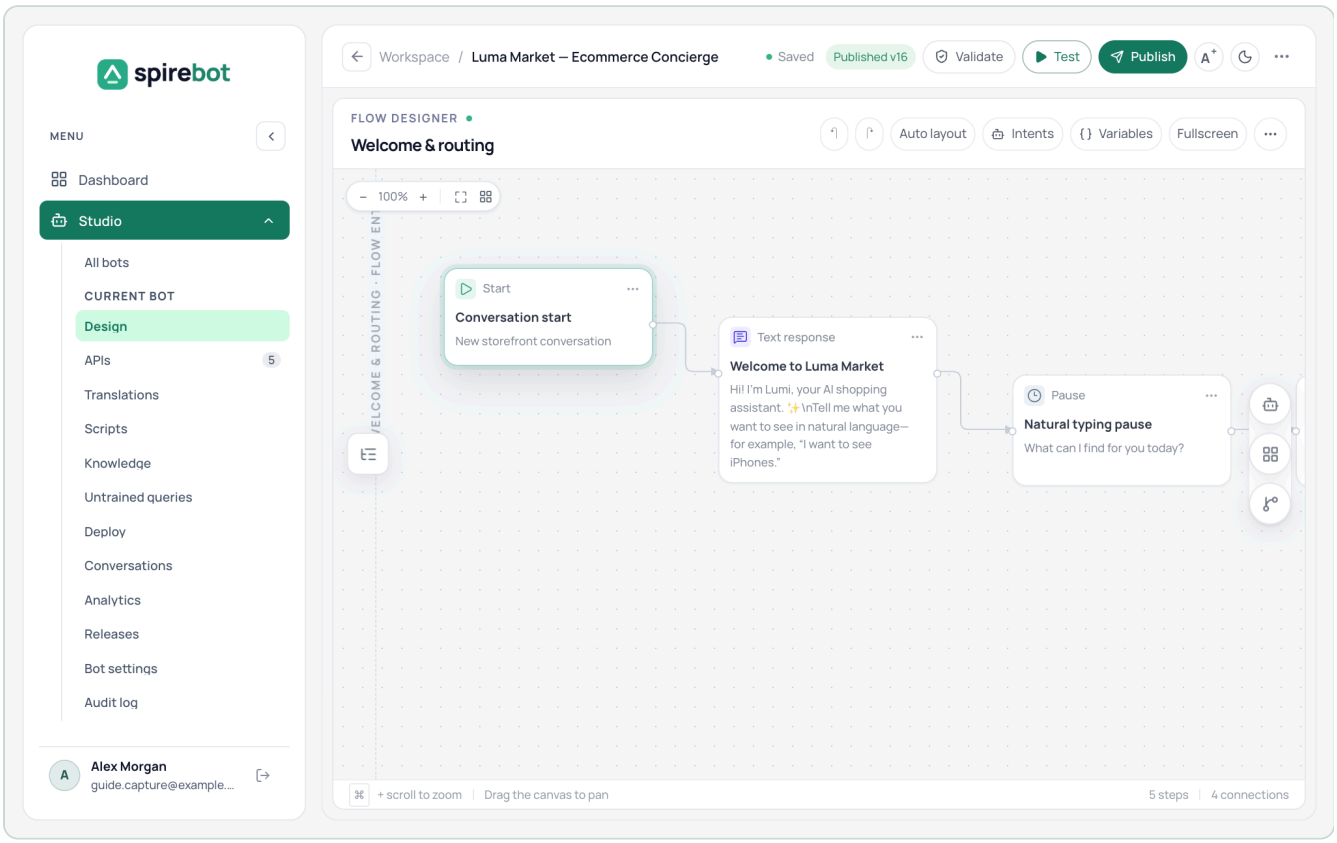


Figure 5. The flow designer for a sample bot.

Area	What it does
Top bar	Shows the bot name, the save status (**Saved**), and the published version. **Validate** checks the draft, **Test** opens the playground, and **Publish** releases the draft.
Flow toolbar	Undo and redo, **Auto layout** to arrange the steps, **Intents**, **Variables**, **Fullscreen**, and more flow options.
Canvas	Scroll to zoom and drag to pan. The zoom controls at the top left include **Fit all steps in view**.
Left edge	Opens the **Flows** panel.
Right edge	Opens the **Spire Bot Assistant**, the **Components** panel, and the **Outline** panel.
Status bar	Shows the number of steps and connections in the flow.

> **IMPORTANT**
>
> **Publish** acts immediately. There is no confirmation step: the saved draft becomes the live version as soon as you select it. Validate and test first.

Flows

A bot is made of several flows, each handling one goal. Open the Flows panel to search flows, switch between them, or create a new one. The flow marked **START** is where every conversation begins.

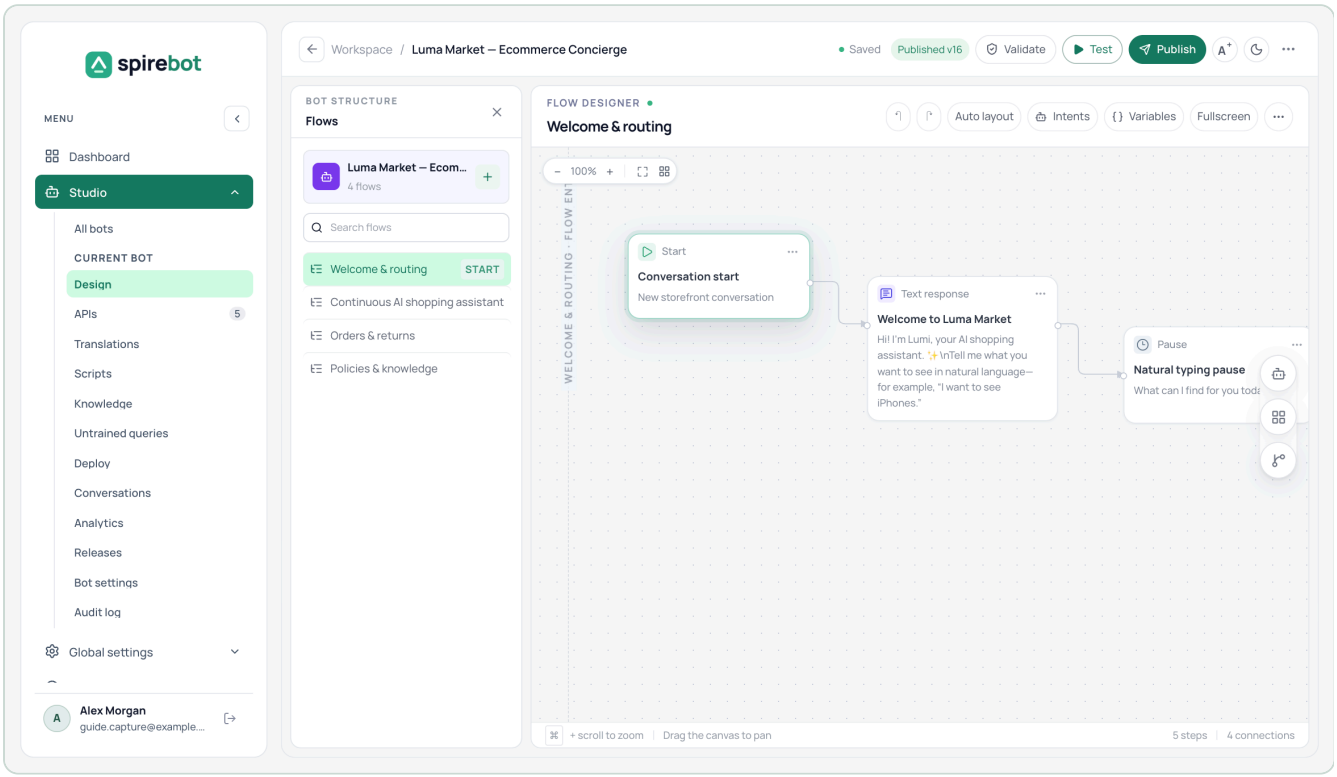


Figure 6. The Flows panel lists every flow in the bot.

Add and connect steps

1. Open the **Components** panel from the right edge of the canvas.
2. Find the step you need. Use the search box if the list is long.
3. Drag the step onto the canvas.
4. Drag from the connector on one step to another step to connect them.
5. Select the step and complete its properties.

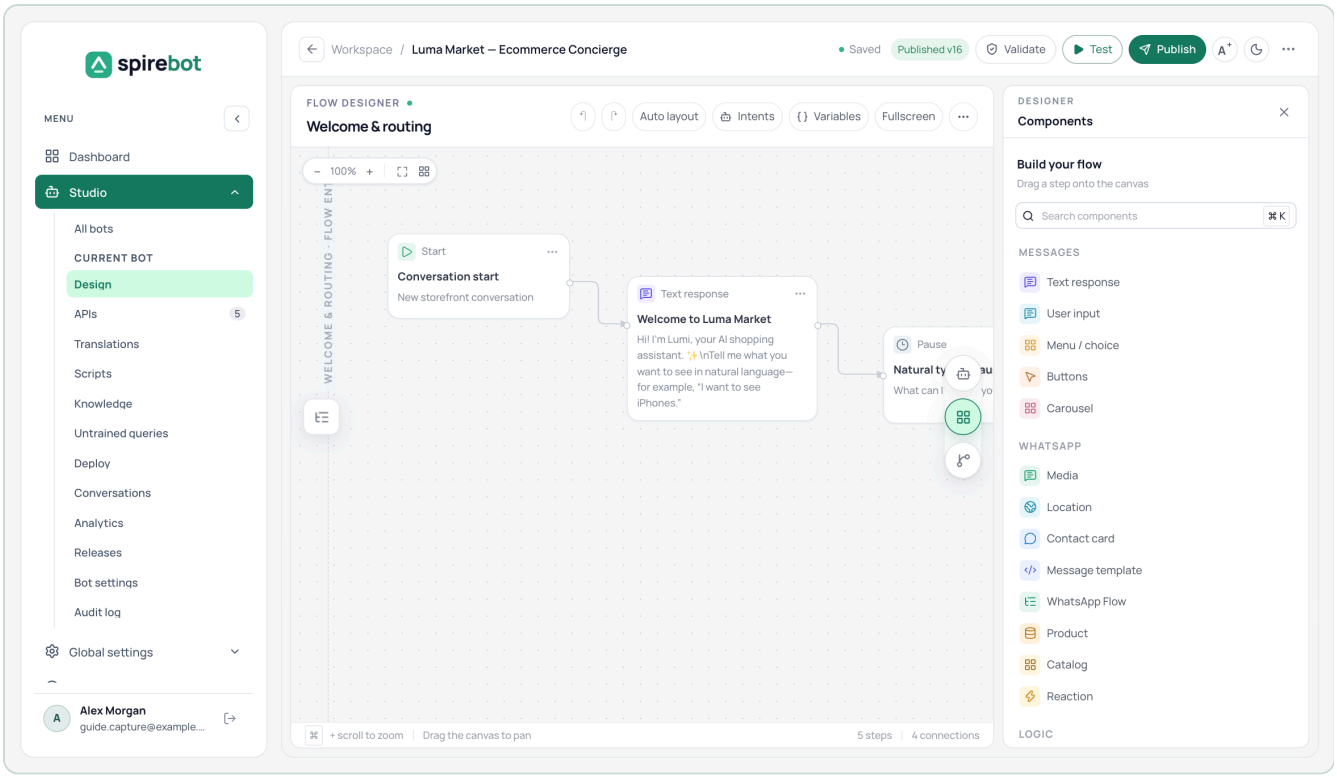


Figure 7. The Components panel.

Group	Steps	Use them to
Conversation	Text response, User input, Menu / choice, Buttons, Carousel, Pause	Send messages and collect answers.
WhatsApp	Media, Location, Contact card, Message template, WhatsApp Flow, Product, Catalog, Reaction	Use formats specific to WhatsApp.
Logic	Condition, AI Router, Set Variable, Jump	Branch on rules, classify natural language, store values, and reuse another flow.
Actions	API Call, Knowledge, Run Script, Live Agent	Call external systems, answer from your documents, transform data, and hand over to a person.

Edit a step

Select a step to open **Step properties**. Give the step a name that describes its outcome, then fill in its content. **Insert variable** adds a placeholder such as `{{customer_id}}` at the cursor. **Flow connections** shows where the step leads next. **Duplicate step** and **Delete step** are at the bottom of the panel.

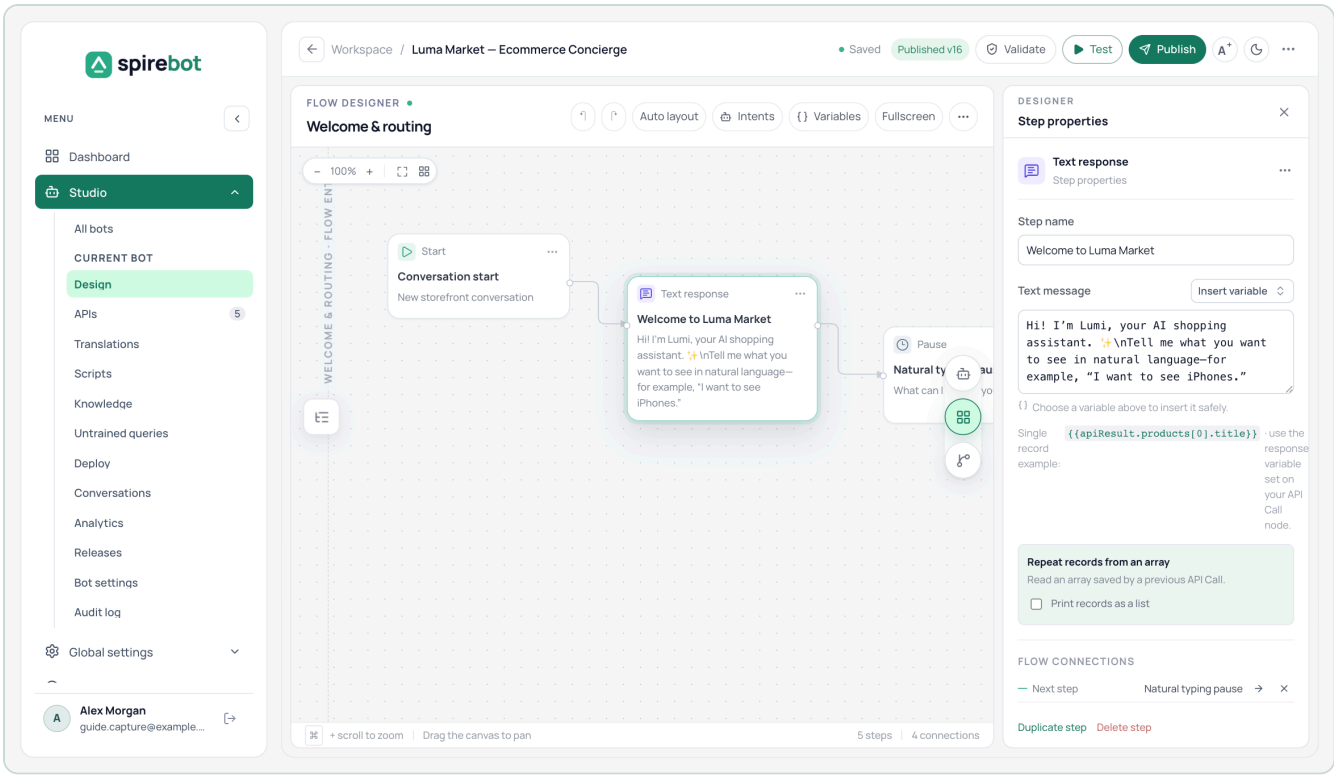


Figure 8. Step properties for a Text response step.

The **Outline** panel lists the steps of the current flow in order, which is a quick way to move around a large flow.

Variables

Variables carry values between steps. Declare each one before a step uses it.

1. Select **Variables** in the flow toolbar.
2. Select **New Variable**.
3. Enter a unique name, choose a type such as Text, Number, or JSON, and choose a scope.
4. Select **Done**.

Use conversation scope for values that must last for the whole session, and turn scope for short-lived values. Use readable names such as `customer_id`, `order_total`, and `needs_handoff`.

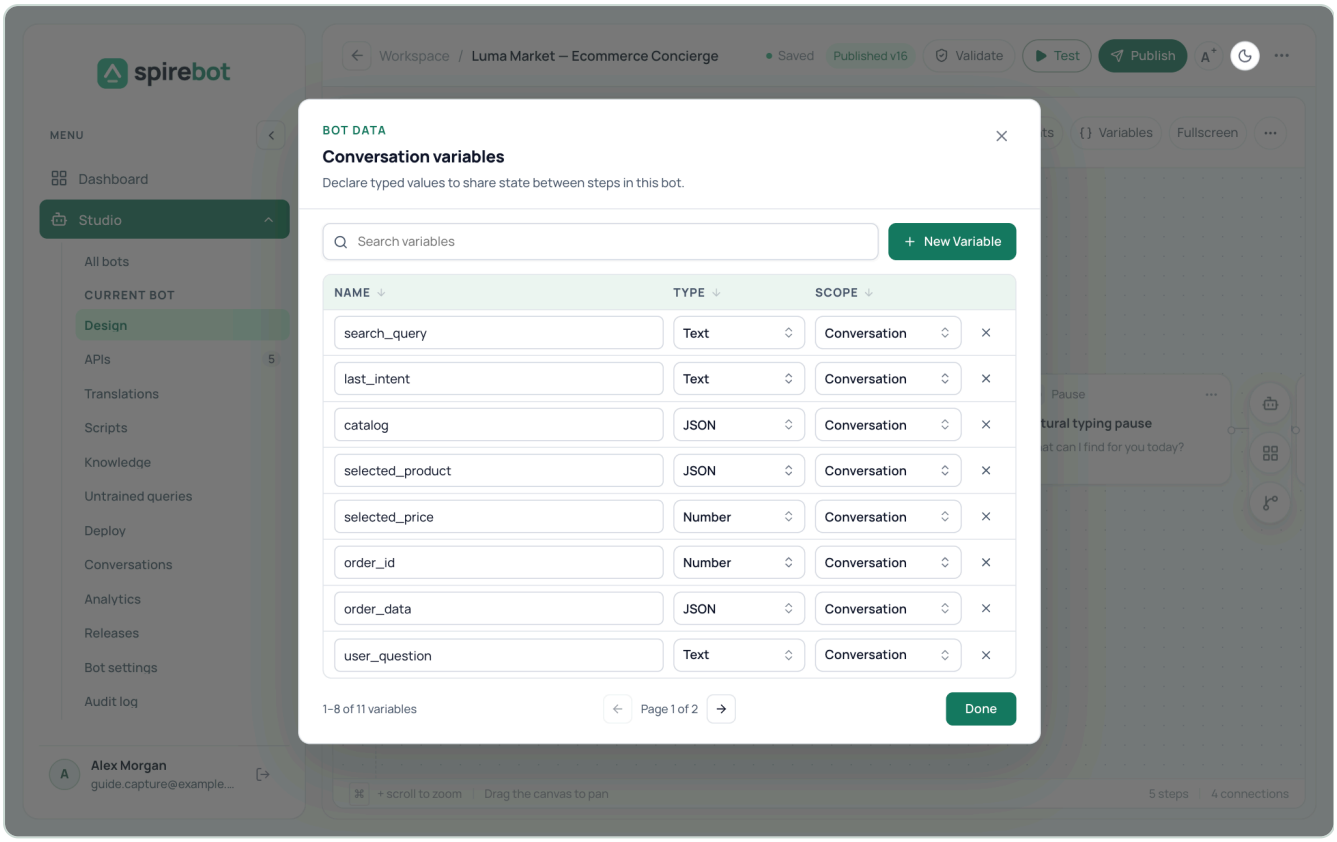


Figure 9. Conversation variables.

Intents

An intent lets a customer reach a flow by typing what they want instead of choosing a menu option.

1. Open the flow and select **Intents** in the flow toolbar.
2. Select **Add intent**.
3. Enter a short name, and describe what belongs in the intent and what does not.
4. Add at least two example messages with different wording, one per line. You can add up to twenty.
5. Under **Details to collect**, add anything the flow needs, such as an order number. The bot takes it from the message or asks for it.
6. Select **Done**.

- Mark a detail as sensitive to keep its value out of traces and transcripts.
- When two intents fit equally well, the bot asks the customer which one they meant.
- A flow can have up to 20 intents. Collected details become conversation variables automatically.

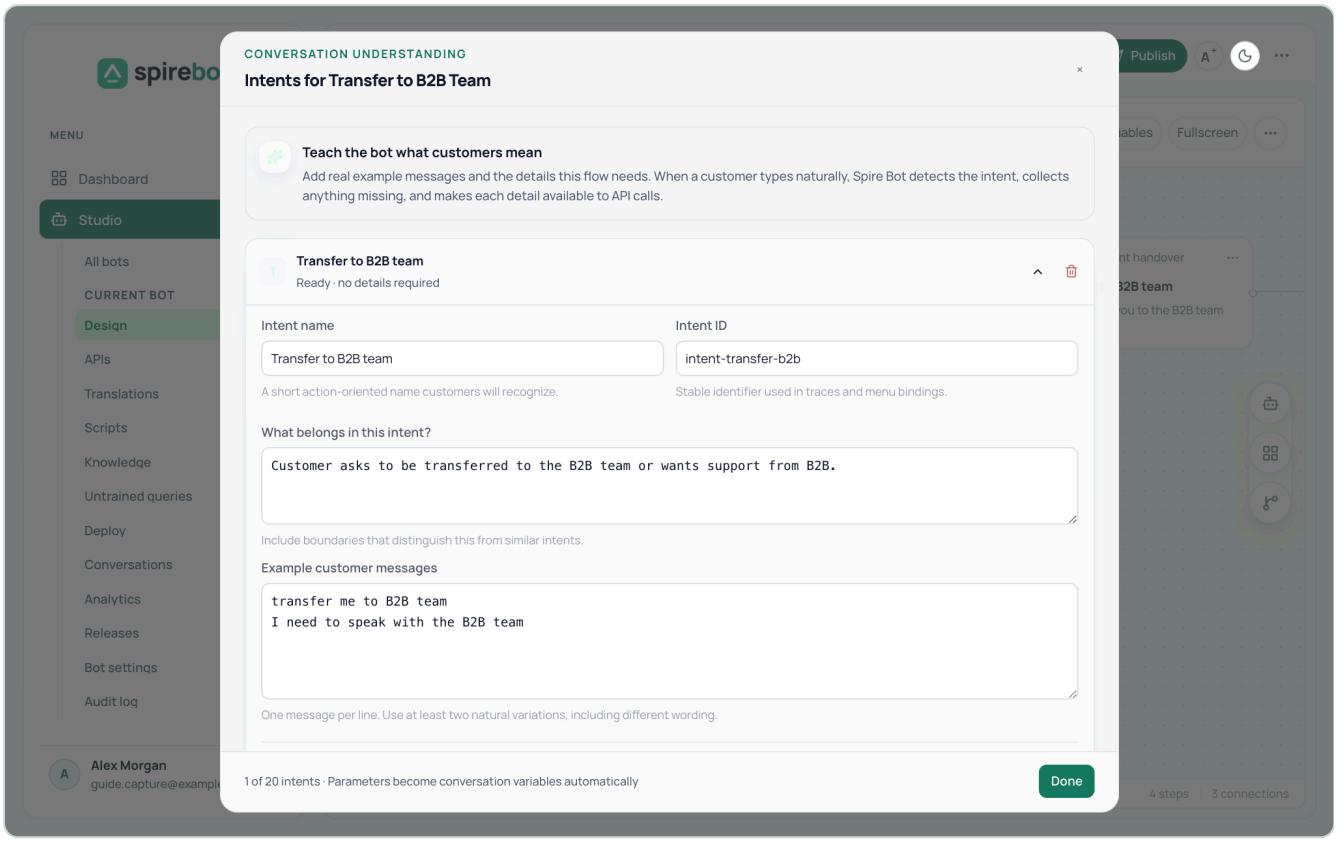


Figure 10. An intent with its description and example messages.

Spire Bot Assistant

The assistant drafts changes for you. Open it from the right edge of the canvas and describe what you want: a new flow, a change to a step, or an API to add. You can also paste a cURL command.

- Start from one of the suggested prompts, or type your own.
- Attach up to 5 Markdown or text files (.md, .txt) with the paperclip, or drop them on the prompt box, to give longer requirements. Files stay attached until you remove them.
- Attached files are context only. Paste a cURL command into the prompt itself to create an API, and keep passwords and keys out of attached files.
- Use **Assistant options** to limit what it may change.
- Review each proposal before you apply it. Proposals change the draft only; publishing stays with you.
- Validate and test the result as you would any flow built by hand.

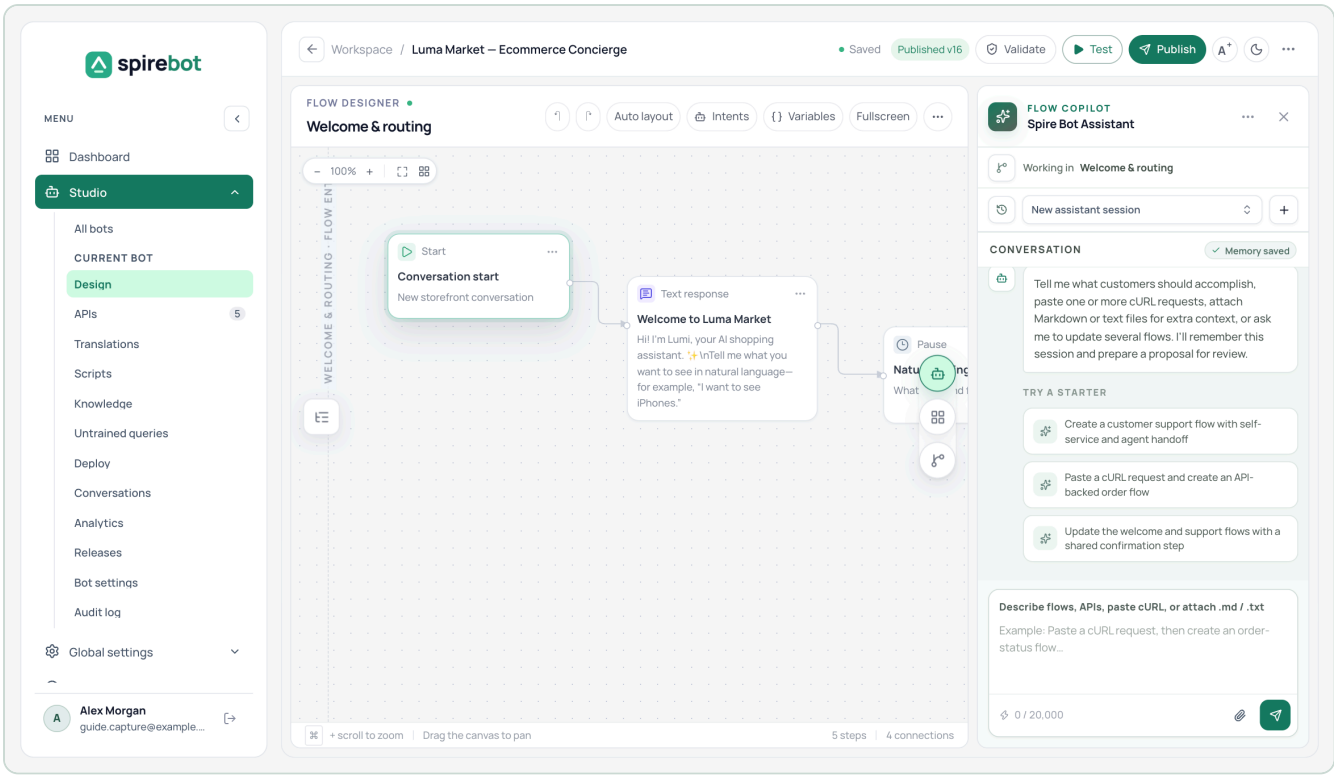


Figure 11. The Spire Bot Assistant beside the canvas.

Design guidelines

Do	Avoid
Name flows and steps by outcome.	Hiding business rules in message text.
Ask one question at a time.	Loops without a retry limit.
Confirm important writes or transactions.	Assuming an API always succeeds.
Give every branch an Else or fallback path.	Collecting data that is not required.
End with a result and the next available action.	Leaving steps disconnected from the entry.

CHAPTER 5

Content and data

Connect the bot to your systems and documents, and prepare its content for every language you serve.

API connections

A connection stores the base URL and credentials for one external service. Connections belong to the workspace, so every bot can use them, and credentials are never copied into a bot.

1. Open **Global settings** → **API connections**.
2. Select **Add connection**.
3. Enter the name, base URL, and authentication details, then save.

- Use the narrowest base URL and the minimum permissions the bot needs.
- Authentication and custom headers are encrypted when stored.
- Select **Edit** to change a connection.

> **NOTE**
> Connections are managed by API Integrators and Administrators. Other roles do not see Global settings.

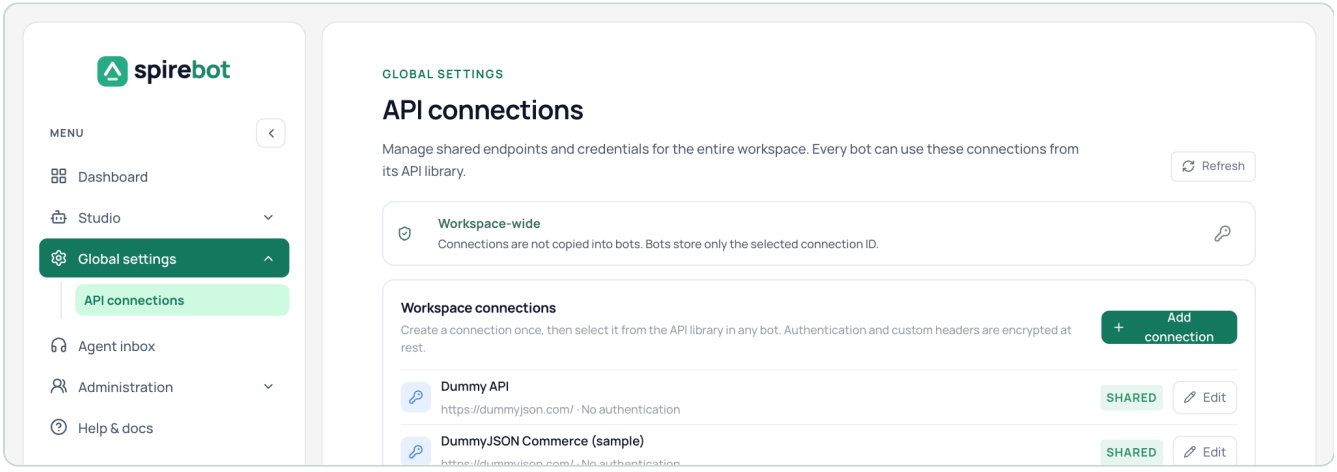


Figure 12. Workspace API connections.

API library

Define each request once in the bot's **APIs** section, then pick it in any API Call step. Changing a saved API updates every step that uses it.

1. Open **APIs** in the bot menu and select **New API**.
2. Enter a name, choose a folder, and choose the connection.

3. Set the method, the path and query, the timeout, and, if needed, the request body and headers. Use **Insert variable** to add conversation values.
4. Test the API with sample values. Check the status and the shape of the response.
5. In the flow, add an API Call step, select the saved API, and connect both its Success and Failure branches.

- **New folder** groups related requests.
- **Import from cURL:** paste a cURL command to fill in a request. Credentials and secret headers are left out on purpose; add them to the workspace connection.
- Map only the response fields the conversation needs.

> **NOTE**
>
> Creating, changing, and testing saved APIs needs the API Integrator or Administrator role. Bot Designers can open the library and use any saved API in an API Call step, but cannot change it. An assistant proposal that adds or changes an API is also refused for a Bot Designer.

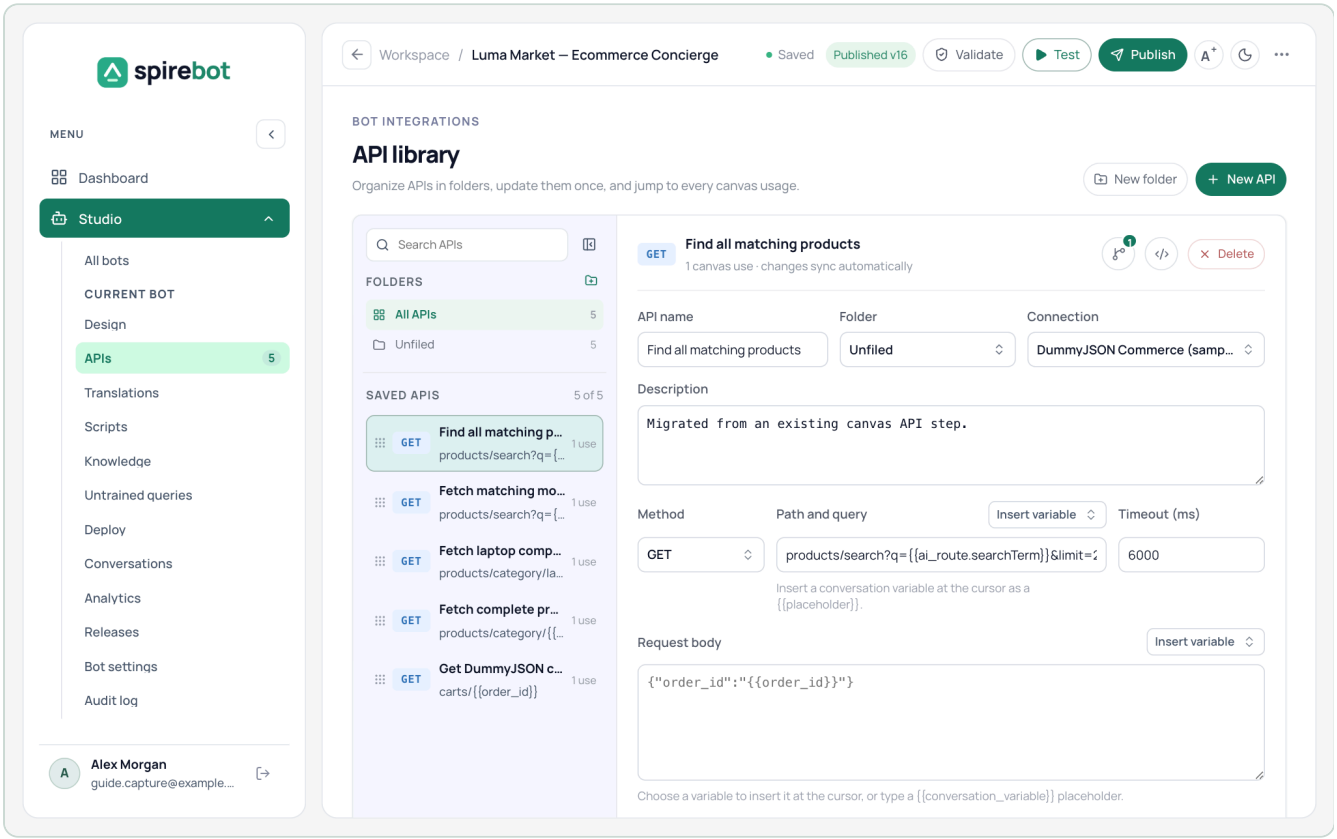


Figure 13. The API library with a saved request open.

Scripts

A script transforms data inside a flow, for example picking one value out of an API response. Open **Scripts** in the bot menu, select **New script**, and write JavaScript in the editor. Then use it in a Run Script step.

- The **Variables** button lists the conversation variables you can read.

- Under **Test script**, enter sample variables as JSON and select **Run test** to see the output.
- Keep scripts small and predictable. Scripts cannot use the network, files, modules, environment variables, dynamic code, or asynchronous work.

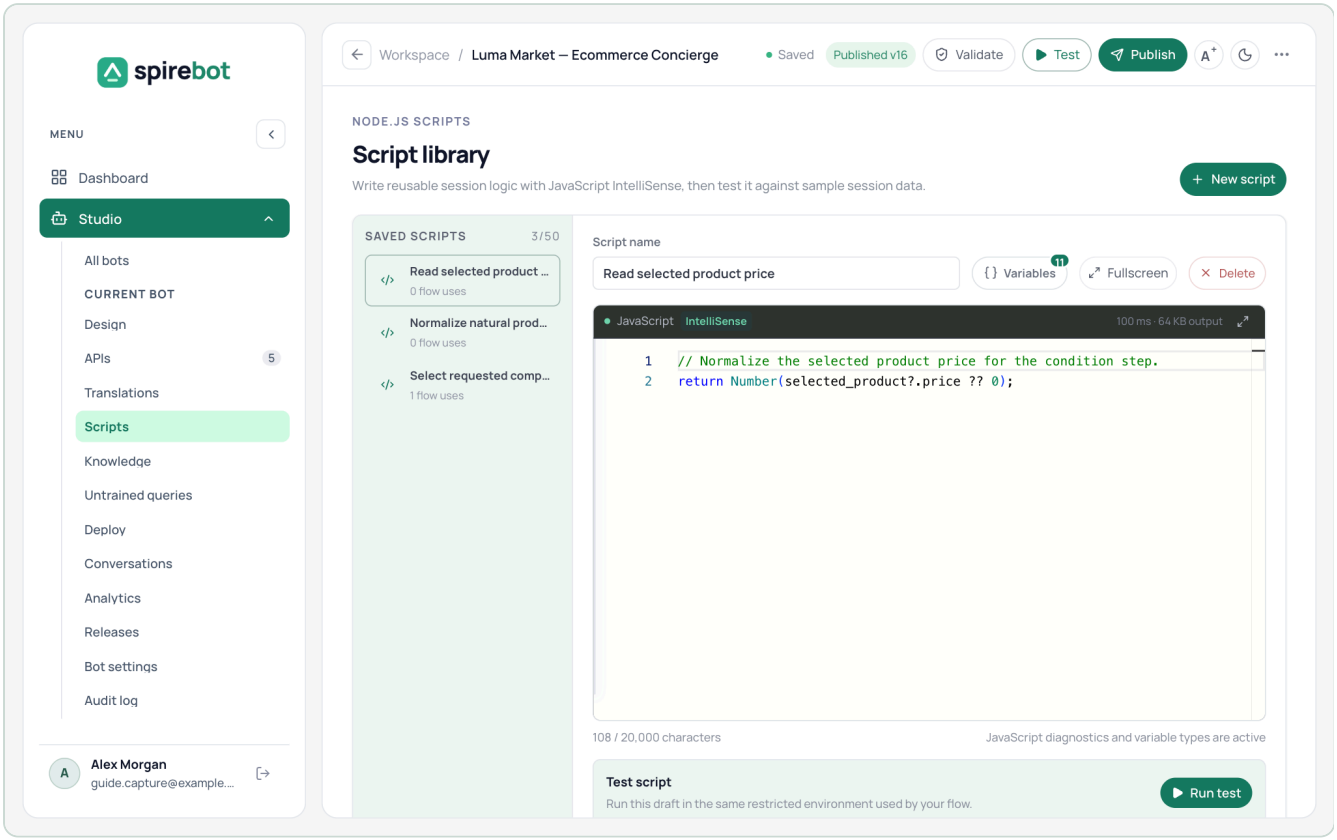


Figure 14. The script library with the editor and test panel.

Knowledge sources

Knowledge sources are the documents the bot answers from in a Knowledge step.

1. Open **Knowledge** in the bot menu and select **Add source**.
2. Upload a document or paste text, and give it a clear title.
3. Wait until the source shows as **ready**.

- Upload focused, current documents. Separate policies by audience, region, or product when needed.
- Write a guarded response for the case where no reliable answer is found.
- Test questions that are answered, ambiguous, and out of scope.
- Replace or remove out-of-date sources as part of content review.

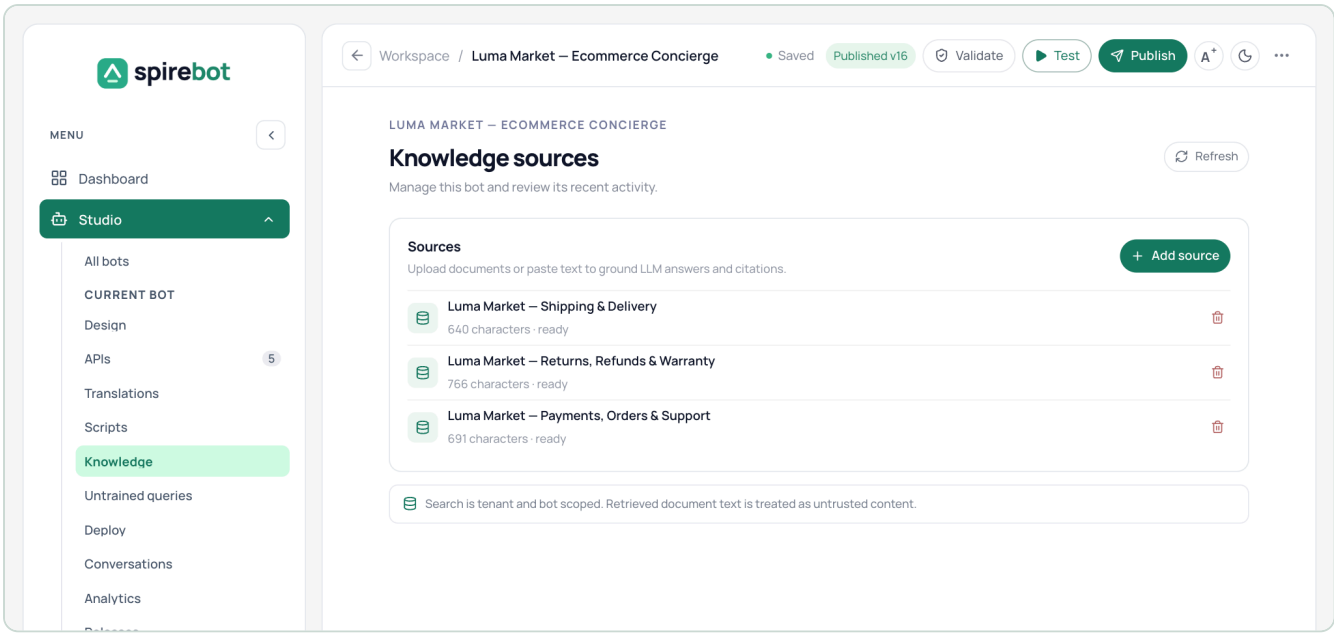


Figure 15. Knowledge sources for a sample bot.

Translations

The **Translations** section lists every customer-facing text in the bot. The source language stays the text you see in the designer.

1. Select **Add language** and choose the language.
2. Enter the translation beside each source text. Use the search box, or the **Missing** and **Needs review** filters, to find what is left.
3. Keep every `{{variable}}` placeholder unchanged in the translation.

- **Detect language** switches the conversation to the customer's language once it is recognised with confidence.
- **Require complete translations** blocks publishing while an enabled language has missing text.

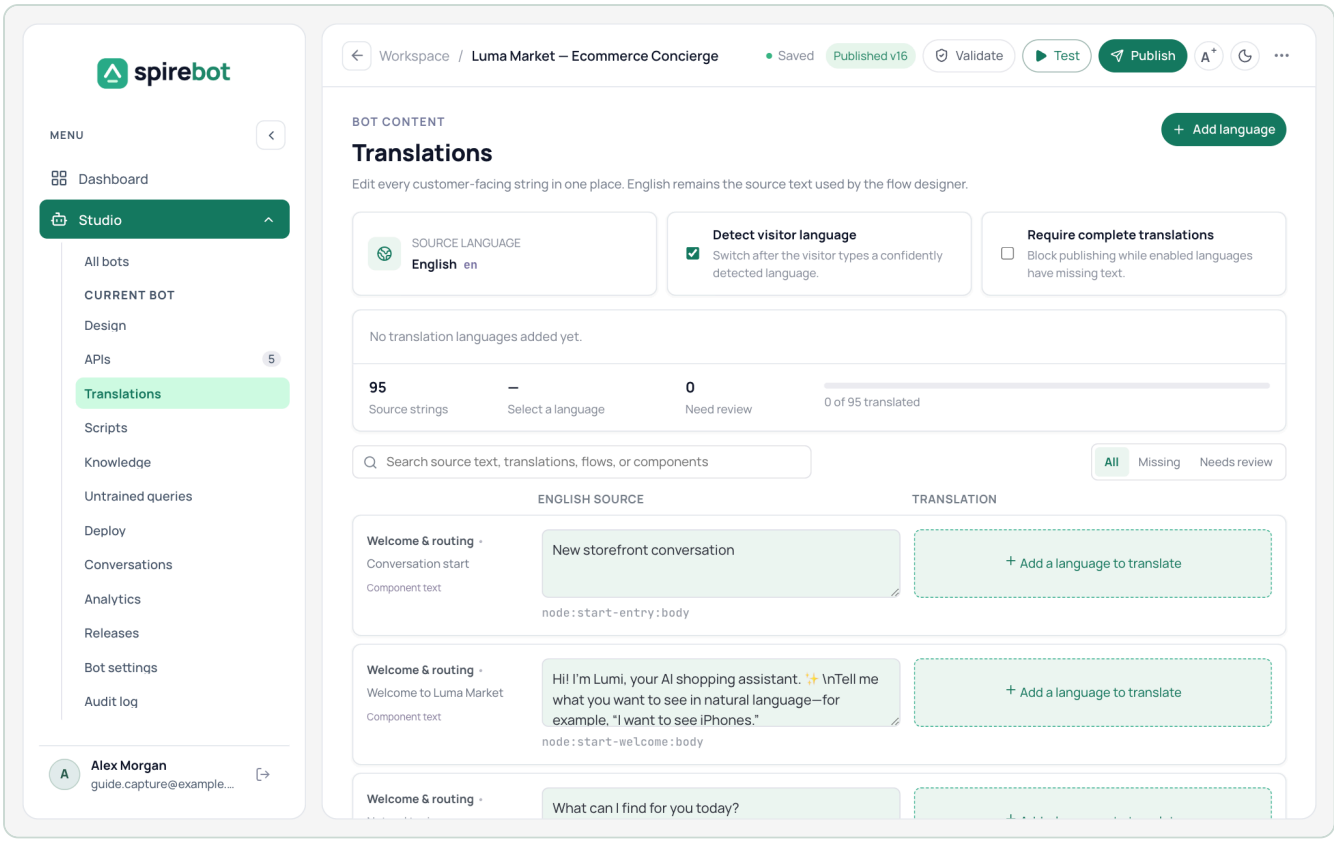


Figure 16. Translations, before a second language is added.

CHAPTER 6

Bot settings

Bot settings control how the bot behaves outside any single flow. Select Save settings after every change.

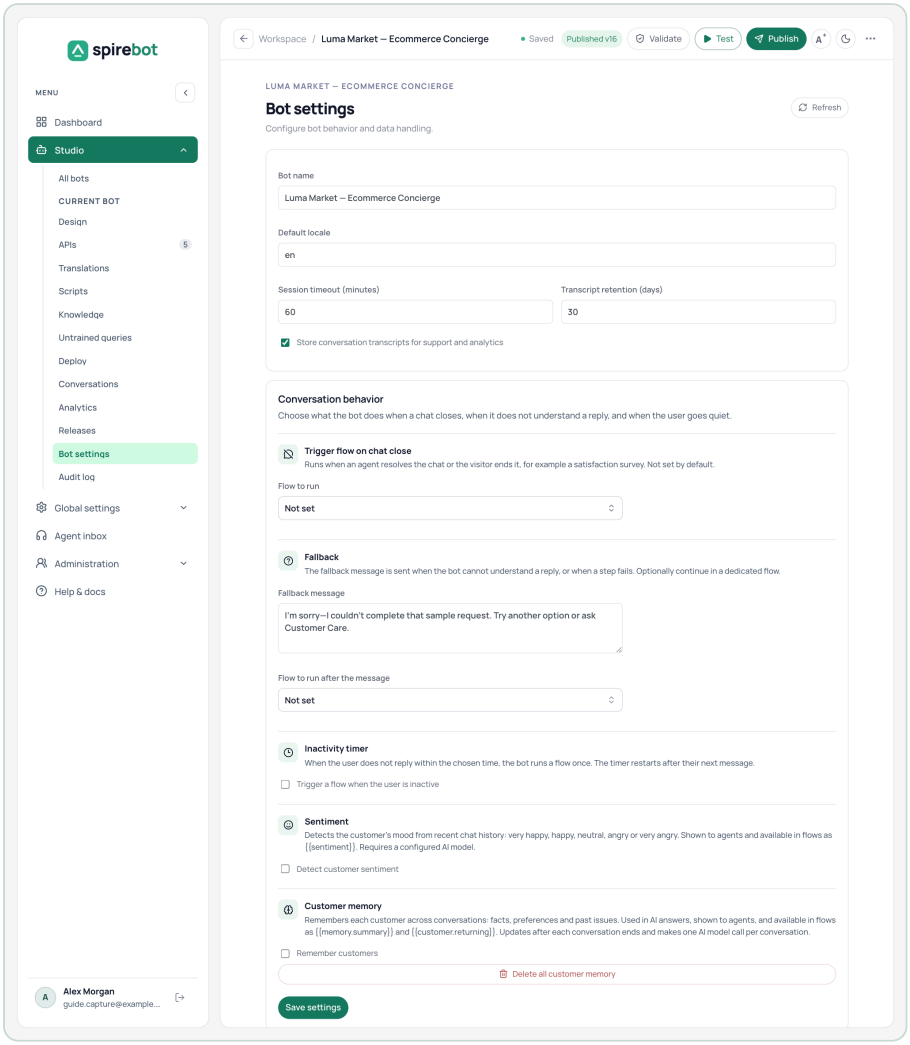


Figure 17. Bot settings and conversation behavior.

General

Setting	Purpose
Bot name	The name shown in the workspace.
Default locale	The language used when no other applies.
Session timeout (minutes)	How long a quiet conversation stays open.
Transcript retention (days)	How long stored transcripts are kept.

Setting	Purpose
Store conversation transcripts	Keeps transcripts for support and analytics. Turning it off also stops the collection of untrained queries.

Conversation behavior

Option	What it does
Trigger flow on chat close	Runs a flow when an agent resolves the chat or the customer ends it, for example a satisfaction survey. It does not run when a flow reaches its own End step or when a session times out.
Fallback	The fallback message is sent when a reply to a menu, buttons, or carousel is not understood, or when a step fails. You can also choose a flow to run after the message.
Inactivity timer	Runs a flow once when the customer has not replied for the chosen time, from 10 seconds to 24 hours. The timer restarts after their next message.
Sentiment	Classifies the customer's mood as very happy, happy, neutral, angry, or very angry. Agents see it in the inbox, and flows can branch on `{{sentiment}}`. It needs a configured AI model.
Customer memory	Remembers each customer across conversations: facts, preferences, and past issues. AI answers use it, agents can see and correct it, and flows can read values such as `{{memory.summary}}` and `{{customer.returning}}`. It needs stored transcripts and a configured AI model. **Delete all customer memory** clears it.

> **IMPORTANT**
> A flow chosen in Bot settings runs only after it is published. If the selected flow exists only in the draft, the settings page warns you.

CHAPTER 7

Validating and testing

Validation checks the structure of the draft. Testing checks how it behaves. Use both before every release.

Validate the draft

Select **Validate** in the top bar. The report lists every issue, such as a missing entry step, a disconnected step, an invalid branch, an undeclared variable, an incomplete translation, or an action that is not configured. Fix each one and validate again until the report says the draft is ready to publish.

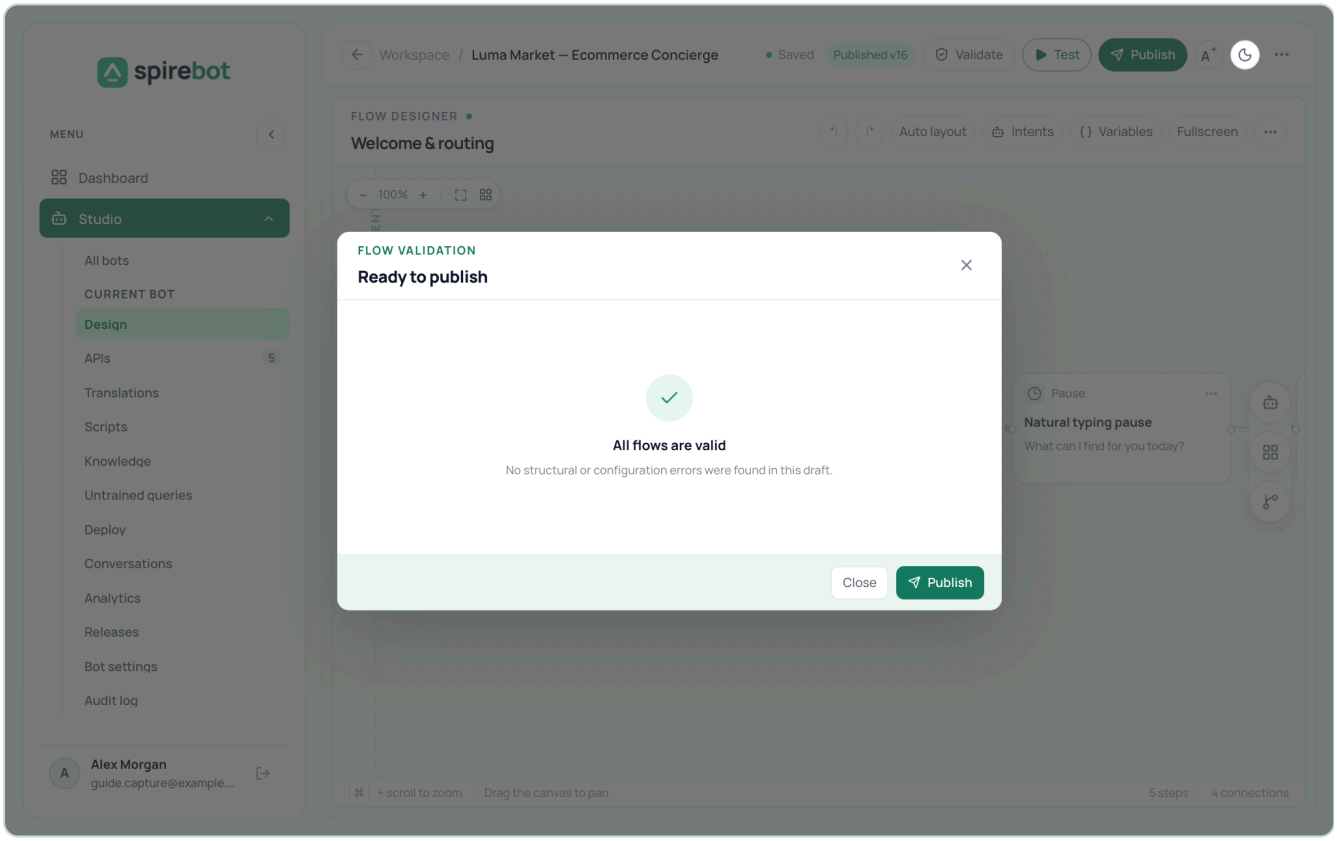


Figure 18. A validation report with no issues.

Test in the draft playground

Select **Test** in the top bar. The playground opens in a new browser tab and runs the draft, not the published version.

- Chat with the bot on the left, exactly as a customer would.
- The **Run trace** on the right lists each step with its status and duration. Select a step to inspect its values. API Call steps also show the request and response.
- **Live API calls** lists the saved APIs the flow can call. When live calls are on, requests go to the real services and may change real data.
- **Reset conversation** starts again from the first step.

Test runs never create a customer conversation, and they are not counted in Untrained queries or Observability.

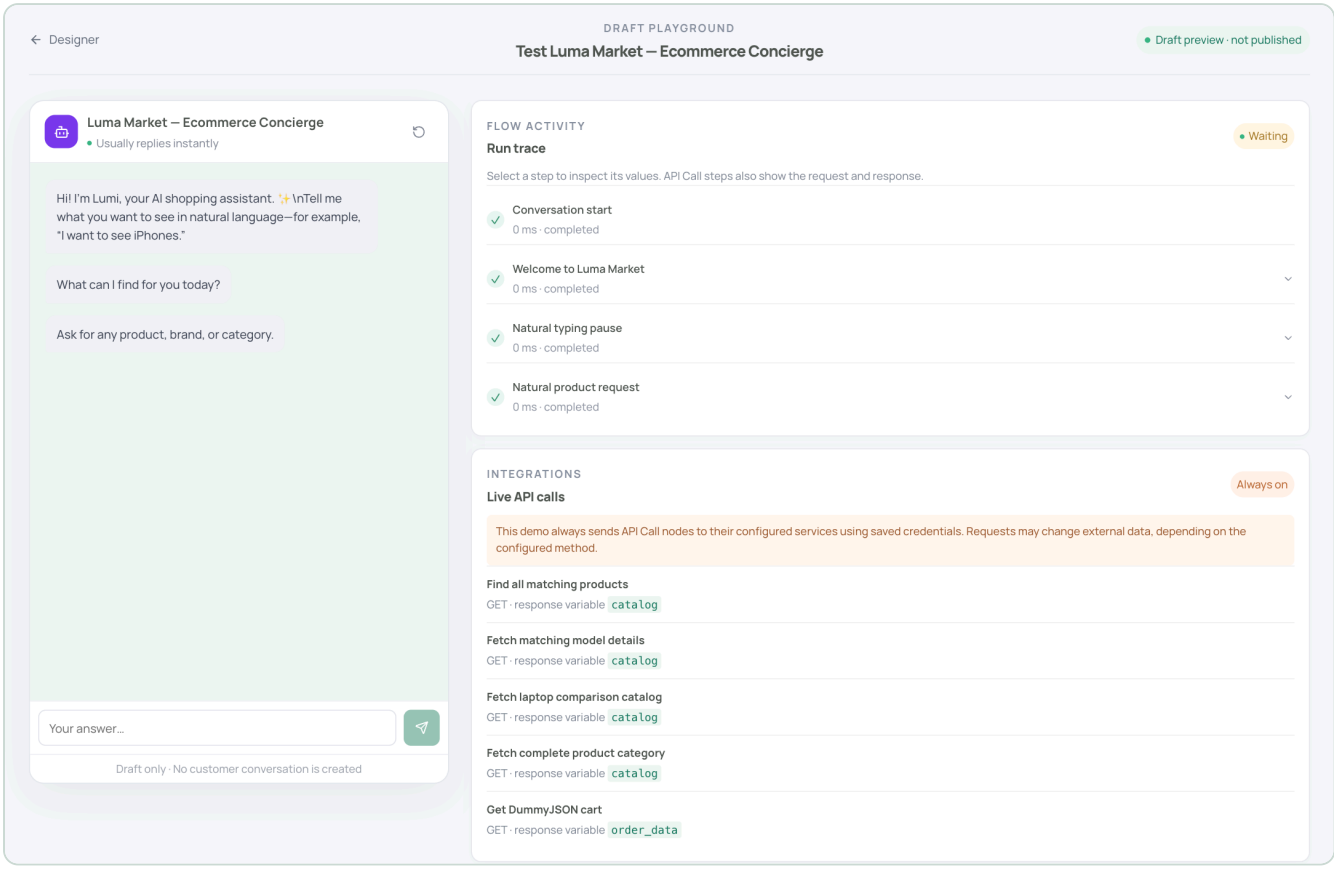


Figure 19. The draft playground with the run trace.

What to test

Case	Check that
Happy path	The customer reaches the intended outcome.
Wrong input	The bot explains the expected format and lets the customer retry.
Empty result	The bot says nothing was found and offers a next step.
Dependency failure	An API failure follows the Failure branch and offers retry or a person.
Fallback	An unrecognised reply gets the fallback message.
Handover	The Live Agent step reaches the right department.
Language	Each enabled language reads correctly.
Channel	The conversation works in the real web widget or WhatsApp number.

IMPORTANT

Never build an endless retry loop. After a few failed attempts, change the interaction: show choices, ask for less, or transfer to a person.

CHAPTER 8

Publishing and deploying

Publishing creates a numbered release from the saved draft. Deploying makes that release available on a channel.

Publish a release

1. Wait for the **Saved** status in the top bar.
2. Select **Validate** and resolve every issue.
3. Test the changed paths and their failure routes.
4. Select **Publish**. The draft becomes the live version at once.
5. Open **Releases** and confirm the new version.

Release history

Releases lists every published version with its date and note. The live version is marked **Active**. Each conversation stays on the version it started with. To return to an earlier version, select **Republish** beside it.

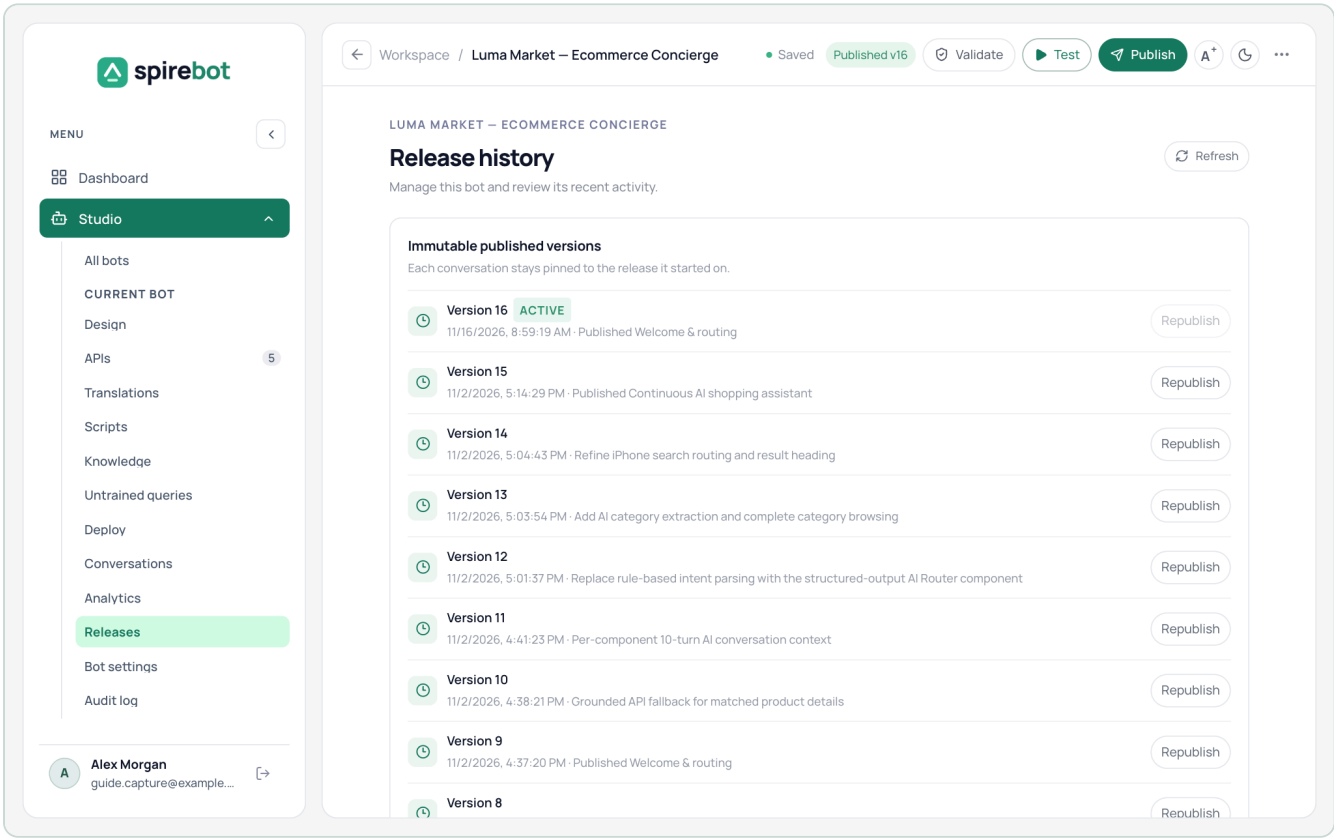


Figure 20. Release history.

Deploy to web chat

1. Open **Deploy** in the bot menu.

2. Under **Allowed website origins**, enter the addresses of the websites that may show the widget, one per line.
3. Select **Create deployment**.
4. Select **Copy embed** and add the script to your website.
5. Select **Test page** to try the widget on a sample page.

Use the switch on a deployment to turn it on or off.

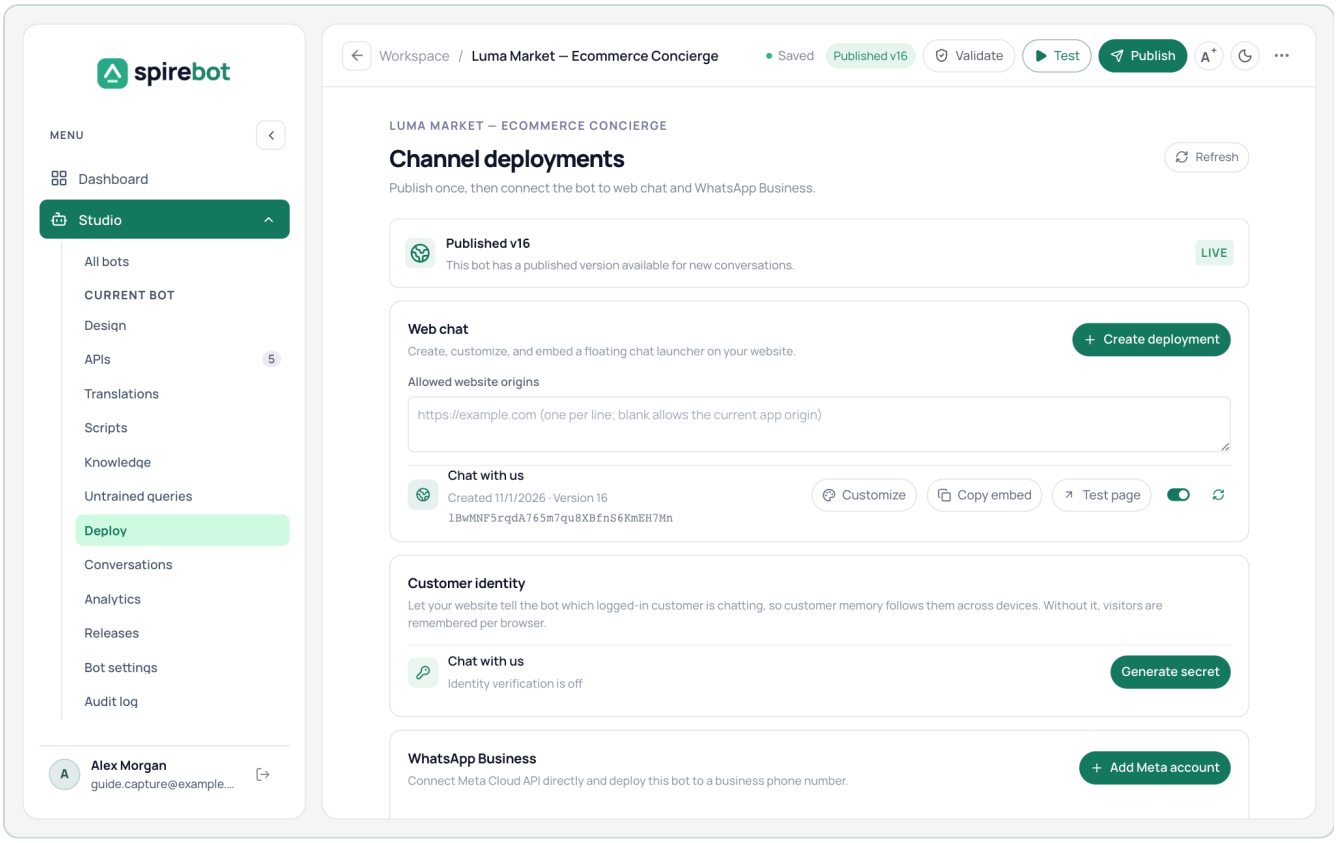


Figure 21. Channel deployments.

Design the web widget

Select **Customize** on a web deployment. No CSS or code is needed, and the live preview updates as you work.

Section	Options
Brand preset	A coordinated palette: Spire Bot, Ocean, Emerald, Sunset, Rose, or Night.
Colors	Primary, Header, Surface, Bot message, and Visitor message.
Header and messages	Widget title, subtitle, message placeholder, and an optional launcher label.
Icons	Separate symbols for the bot and the launcher.
Layout	Light or dark theme, position at the bottom right or bottom left, size, and shape.

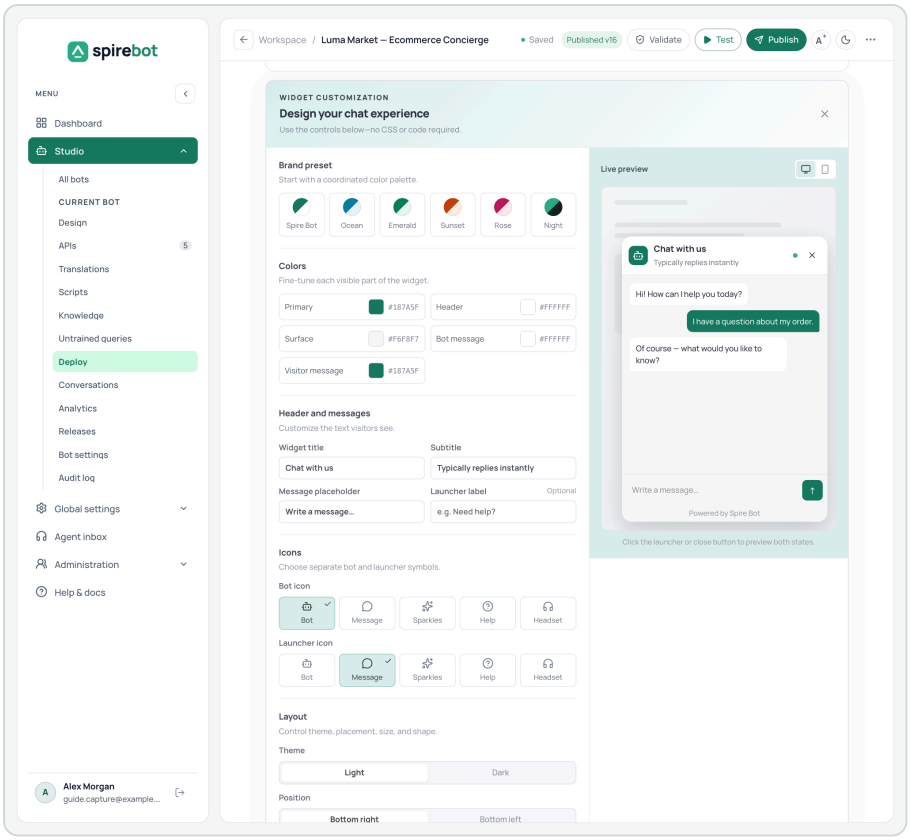


Figure 22. The widget designer with its live preview.

Customer identity

By default, web visitors are remembered per browser. To let customer memory follow a signed-in customer across devices, select **Generate secret** under Customer identity and have your website pass a signed customer ID to the widget.

Deploy to WhatsApp Business

Under **WhatsApp Business**, select **Add Meta account** to connect your Meta account, then choose the business phone number to deploy to. Confirm templates, media, catalog assets, and channel limits before you publish.

After every release

- Run a live test from the real channel: the start, one core outcome, one failure path, and a handover.
- Watch the first conversations and failures closely.
- Keep the previous version available so you can republish it if needed.

CHAPTER 9

Live support

A Live Agent step places the conversation in a department queue. These tools help agents pick it up, answer quickly, and leave a useful record.

Users and agents

Open **Administration** → **Users & agents**. The directory lists every account with its roles, departments, and concurrent chat limit. Administrators and Security Admins control access from this page.

- Select **Create user** to add an account.
- Open a user's row to change their roles, departments, or chat limit.
- Each agent can handle 1 to 100 chats at once. The default is 5.

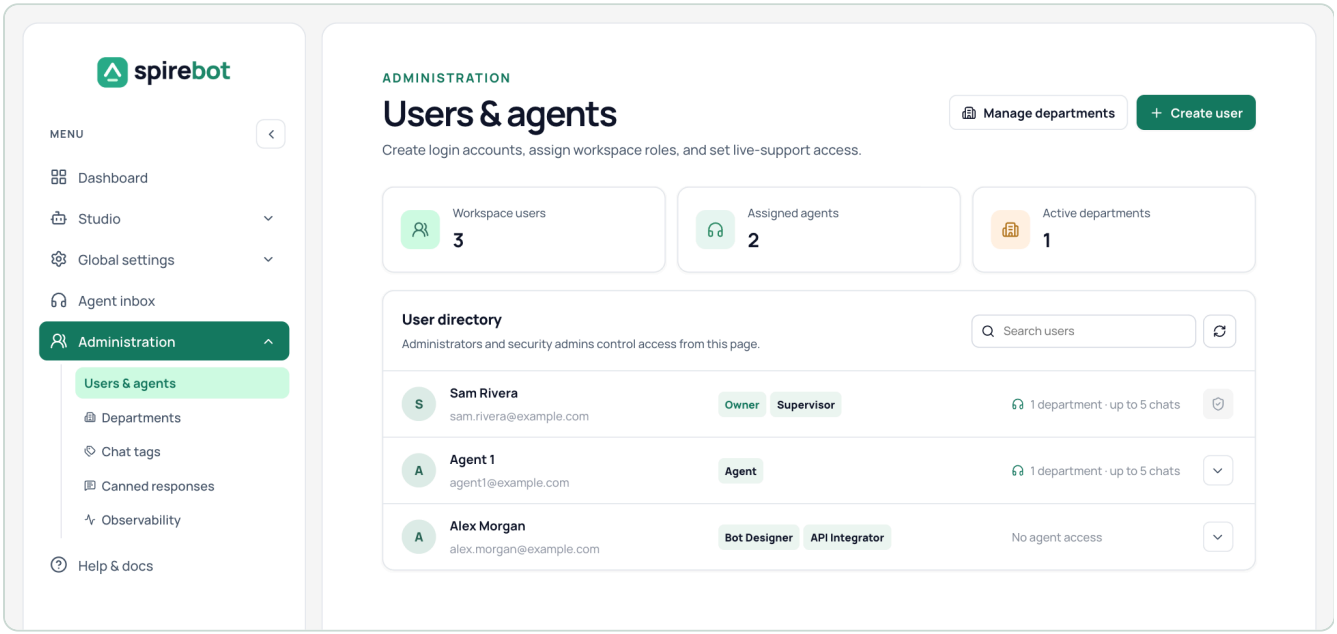


Figure 23. Users and agents.

Roles

Under **Workspace roles**, tick every role the person needs. A user can hold several roles, and their access is the sum of them. Leave all roles unticked for someone who only works in the Agent inbox.

Role	What it allows
Administrator	Everything: every bot, setting, user, department, and the Observability page.
Bot Designer	The whole Studio: design, translations, scripts, knowledge, untrained queries, testing, export, deploy and publish, conversations, analytics, and bot settings. API configurations are read-only.

Role	What it allows
API Integrator	Global settings (API connections and WhatsApp accounts) and each bot's **APIs** section, including API tests. The rest of the bot is read-only.
Security Admin	Users & agents and the audit log: create, edit, and remove users, assign roles, and set department access.
Analyst	Read-only access to bots, conversations, analytics, and untrained queries.
Agent	Set per department. Works the Agent inbox for those departments.
Supervisor	Set per department. Everything an Agent does, plus the team view, chat tags, and canned responses for those departments.

- Give one person **Bot Designer** and **API Integrator** together if they both build flows and configure APIs.
- Only an Administrator can grant the Administrator role or change an administrator's account. A Security Admin cannot.
- The workspace **Owner** is the account the workspace was set up with. It always has full access and cannot be edited or removed.
- A role change takes effect the next time the user loads a page. They do not need to sign in again.
- Importing a bot replaces its API configurations, so it needs Bot Designer and API Integrator together, or Administrator.

Departments

A department is a group of agents that receives handovers. Open **Administration** → **Departments**.

1. Enter a name and description, then select **Create department**.
2. Select the department in the list.
3. Give each member the **Agent** or **Supervisor** role.
4. Select **Save roles**.

Only Administrators create and edit departments. Security Admins can still place a user in an existing department from **Users & agents**.

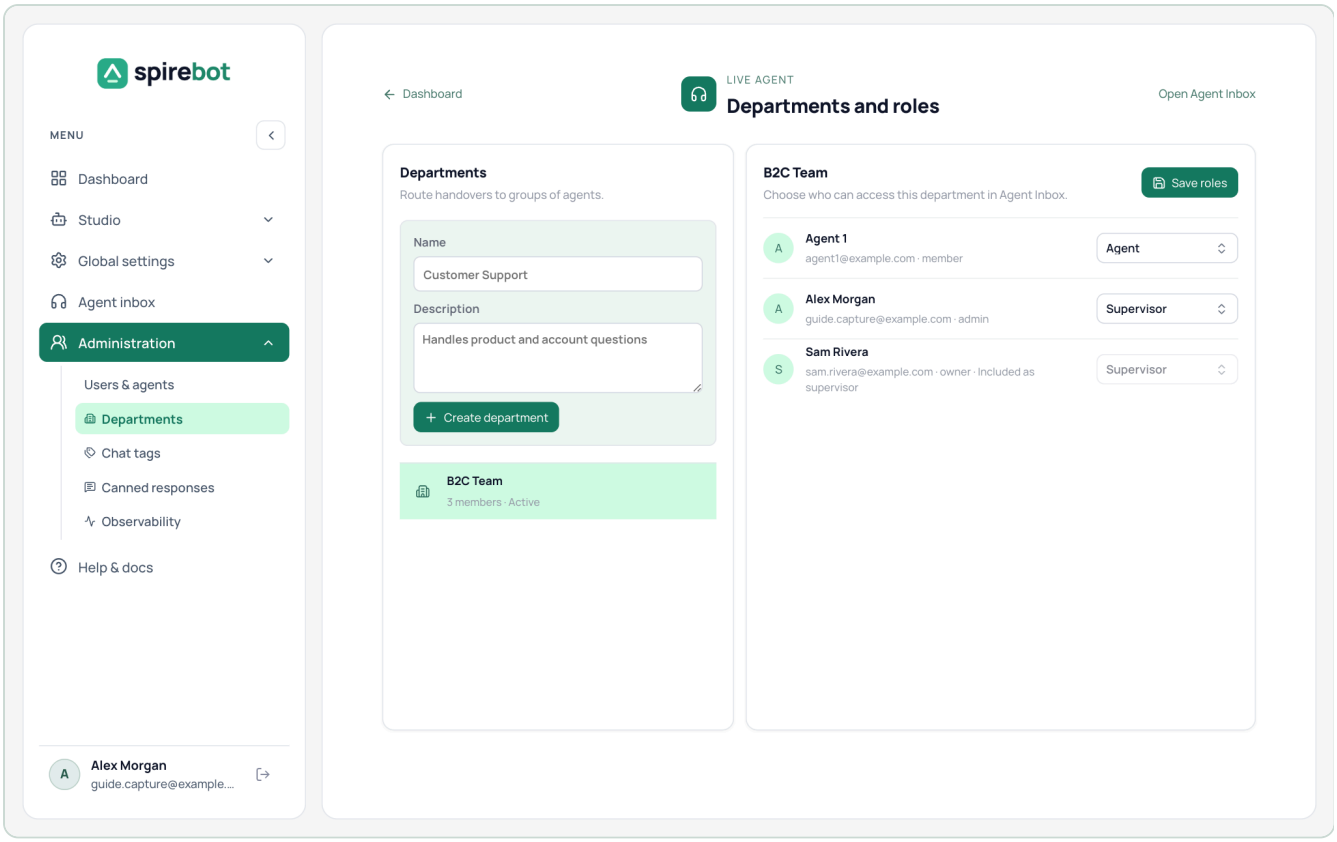


Figure 24. A department and the roles of its members.

The Agent inbox

You need an Agent or Supervisor role in at least one department to open the Agent inbox.

View	Shows
Queue	Handovers waiting for an agent in your departments.
My chats	The chats you have claimed, with your current load.
History	Resolved conversations and their tags.
Team	The agents in your departments.

1. Open **Queue** and select a waiting conversation to read its transcript.
2. Select **Claim** to take it. Claim is disabled when you have reached your chat limit.
3. Reply to the customer. Type `/` to insert a canned response.
4. Select **Resolve** when the issue is closed, then choose tags in the picker that opens.

- The bell button turns on desktop alerts for new handovers.
- Your load, for example “2 of 5 chats”, is shown at the top of the inbox.
- When sentiment and customer memory are on, the inbox shows the customer’s mood and what the bot remembers about them.

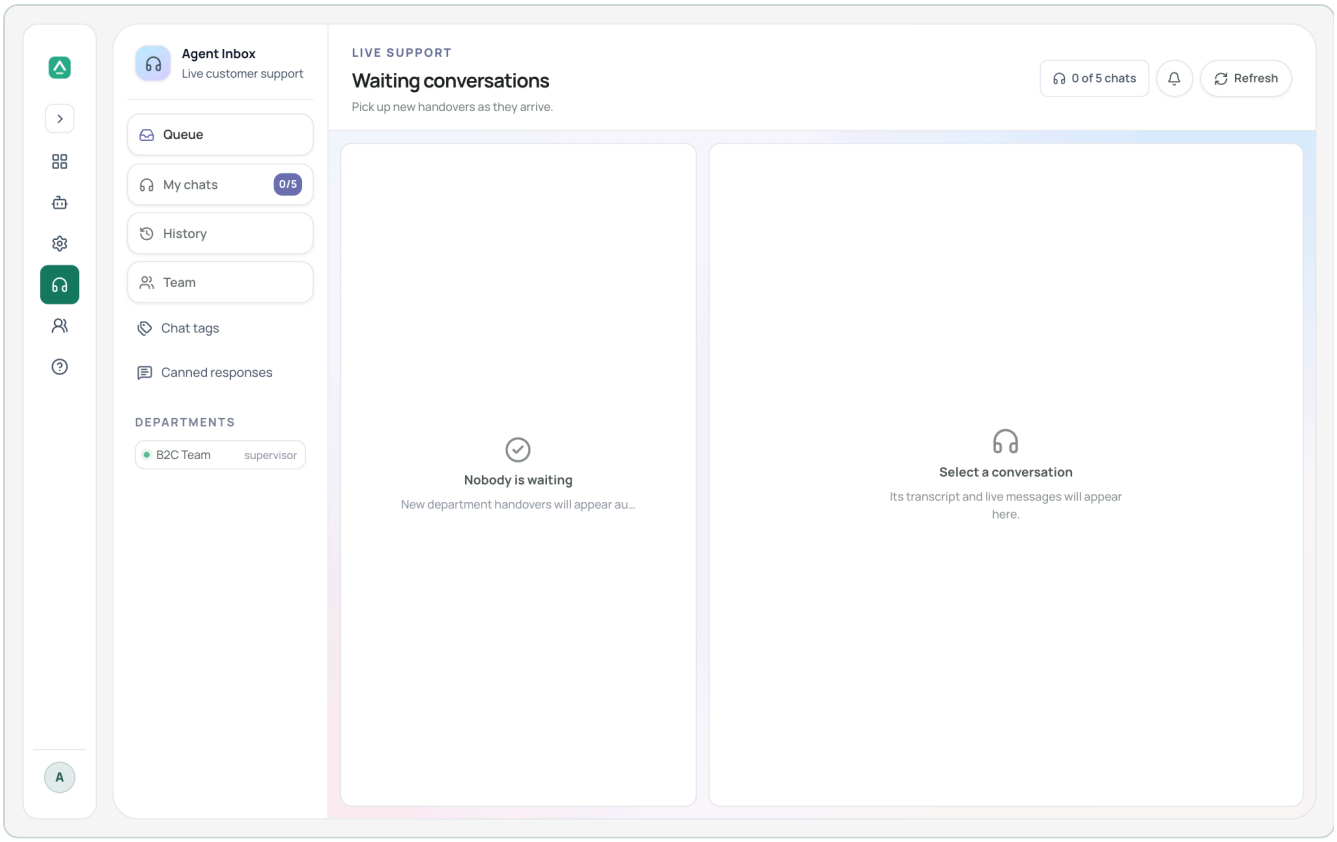


Figure 25. The Agent inbox with an empty queue.

Chat tags

Supervisors and Administrators manage tags under **Administration → Chat tags**. Create a category, such as Issue type or Outcome, then add tags to it. The page shows how many chats use each tag.

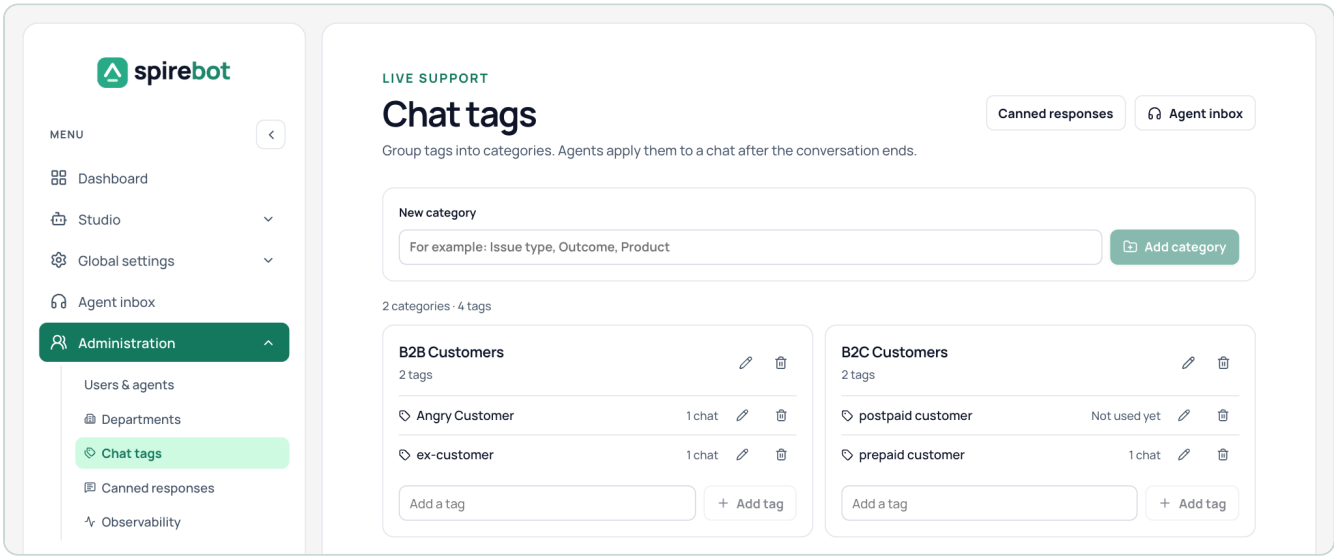


Figure 26. Tag categories and their tags.

Canned responses

Saved replies are managed per department under **Administration → Canned responses**. Select **New response** to add one. In a chat, type `/` to filter by shortcut, use the arrow keys to choose, press Enter to

insert, and Enter again to send.

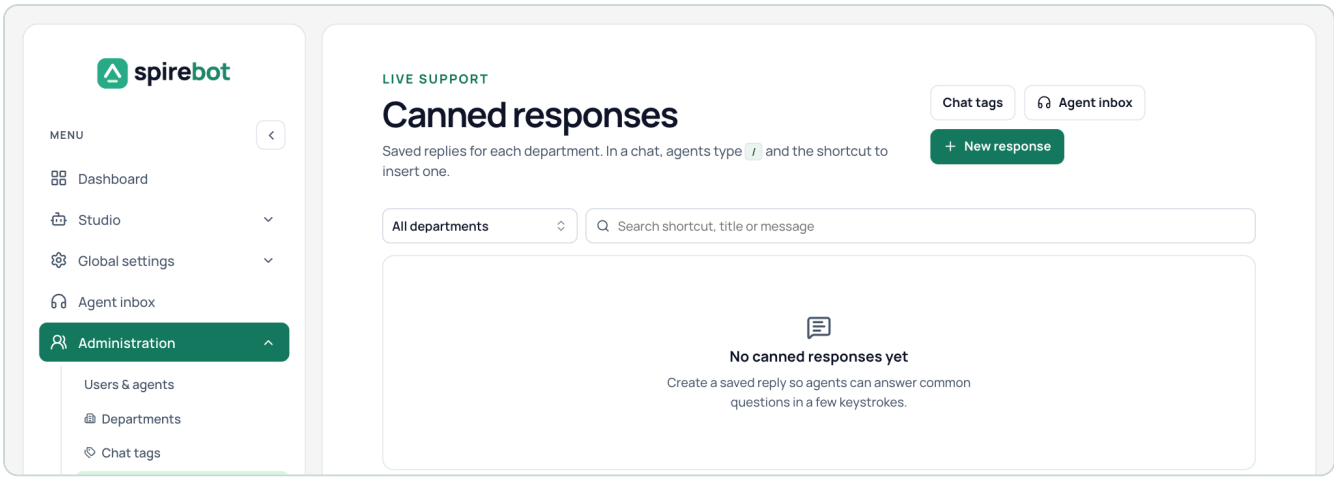


Figure 27. Canned responses, before any are saved.

> **NOTE**
>
> Customers can end a chat themselves with the **End chat** button in the web widget. Ending a chat, or an agent resolving it, runs the close flow if one is set in Bot settings.

CHAPTER 10

Operating and improving

Use what happens in production to decide what to fix next.

Conversations

Conversations in the bot menu lists recent conversations with their version, channel, time, and outcome. Filter by outcome, and select a conversation to see the trace of the steps it ran. Sensitive values are removed from the trace.

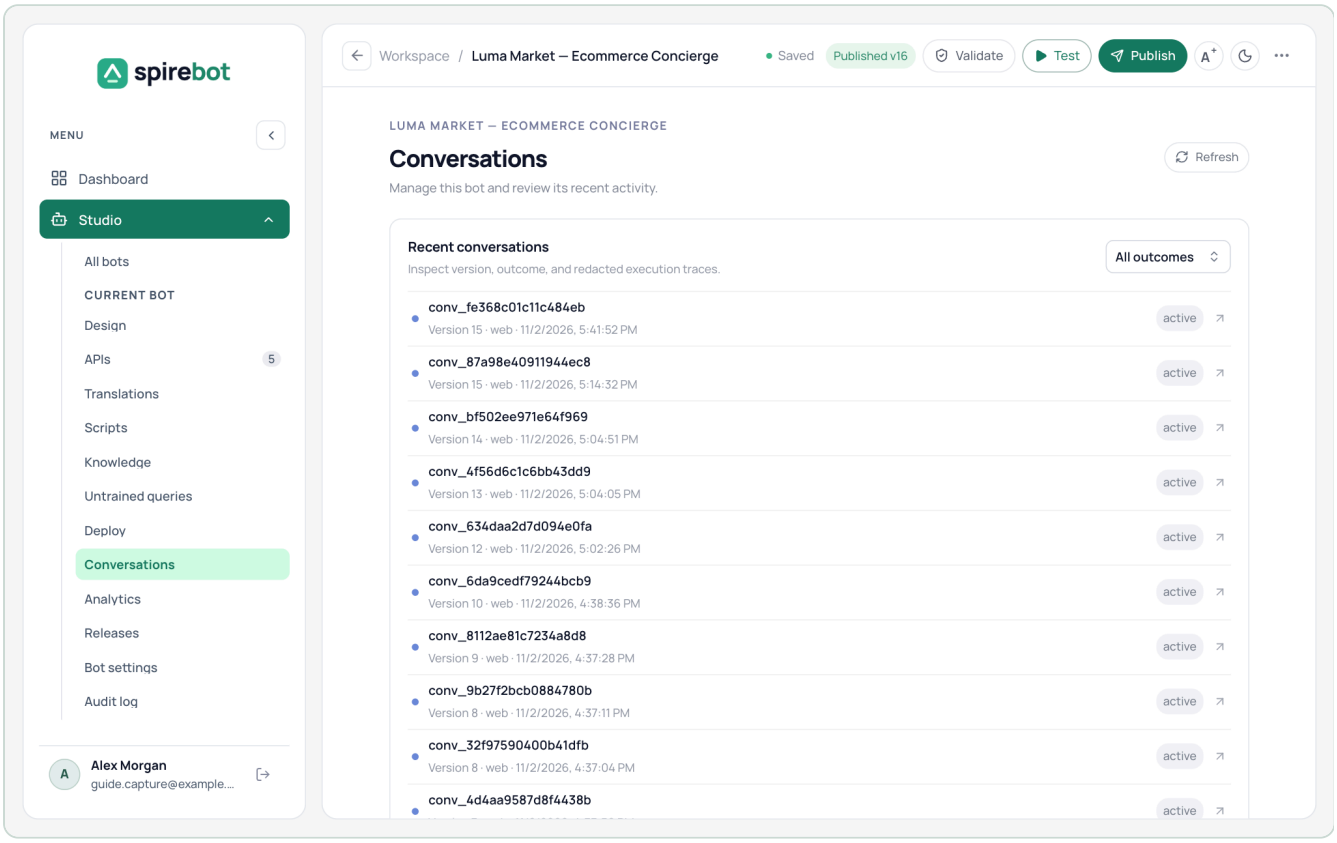


Figure 28. Recent conversations for a bot.

Analytics

Analytics shows the number of conversations, how many completed, the completion rate, how many failed, and how many unanswered questions are open, with daily volume for the last 30 days.

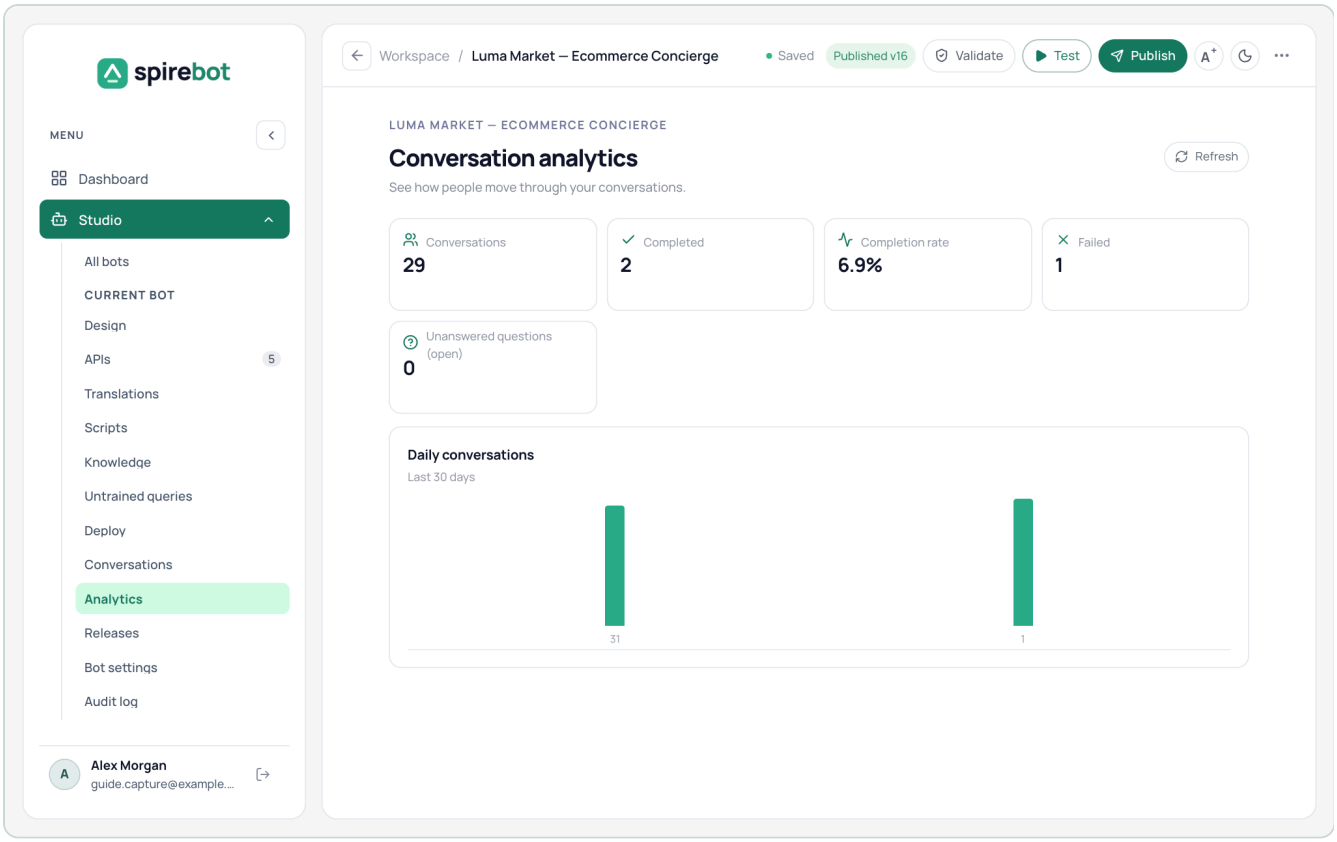


Figure 29. Conversation analytics.

Untrained queries

A question appears under **Untrained queries** when the knowledge search finds nothing, when an answer could not be verified against the sources, when an API returned no data, or when a reply was not understood.

- Repeated questions are grouped and sorted by how often they were asked. Start at the top.
- **Add answer** saves a question and answer to knowledge. It is live at once.
- **Add to intent** adds the customer's wording as an example of an existing intent. It takes effect after you publish.
- **Ignore** removes spam and out-of-scope questions from the open list.
- A trained question that fails again returns to the open list.
- Administrators and Bot Designers can train the bot. API Integrators and Analysts can review the list.

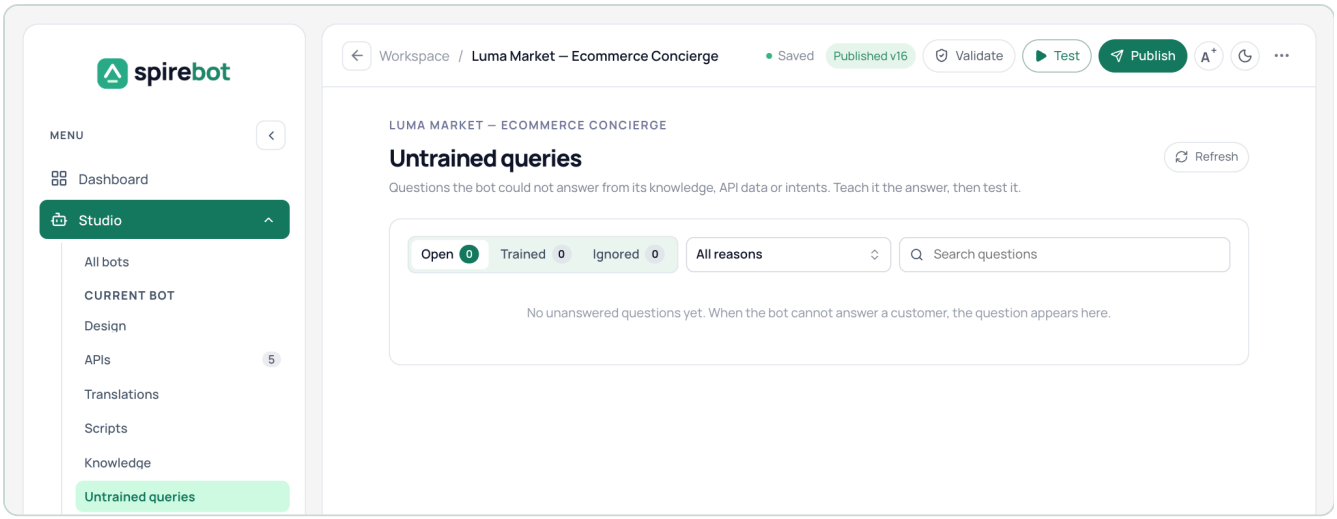


Figure 30. Untrained queries, with no open questions.

Observability

Administrators open **Administration → Observability** for the health of the whole workspace. The page refreshes every 30 seconds.

- **Service health** shows the database, cache, AI model, event delivery, and the background workers as Healthy, Down, or Unknown.
- **Errors** can be filtered by period and by bot. Select a frequent error to see only its occurrences.
- Error history is kept for 30 days and never contains customer text or credentials.

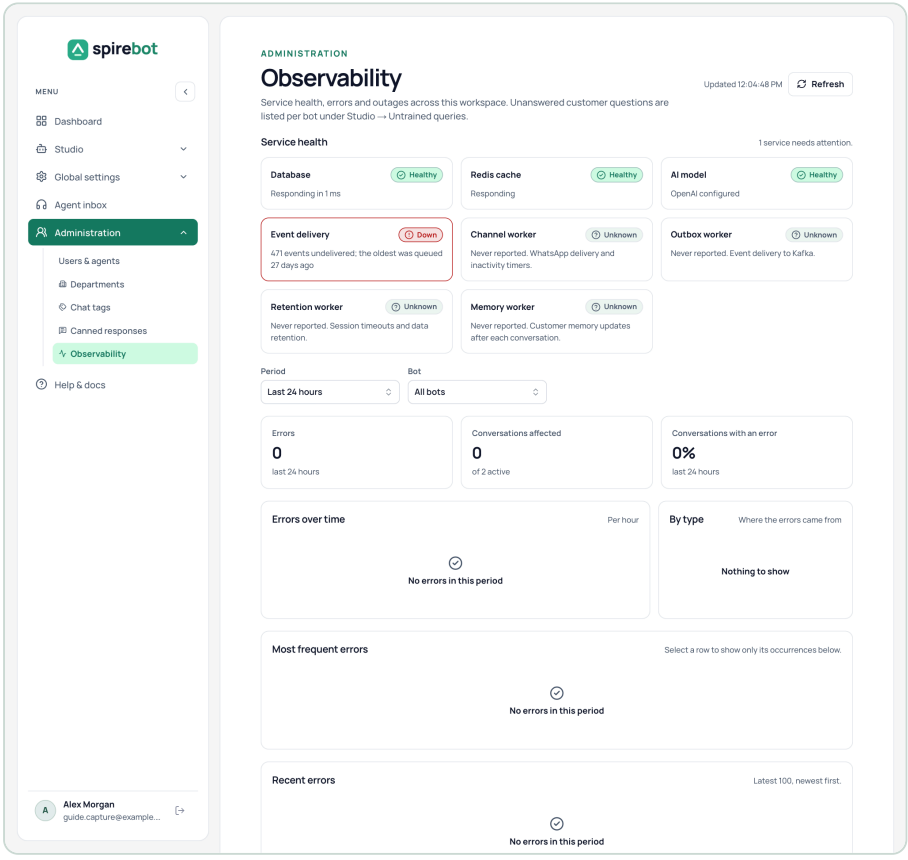


Figure 31. Observability. In this example, event delivery needs attention.

> **NOTE**
>
> Unanswered questions and outages are kept apart. A question the bot could not answer is a training gap and goes to Untrained queries. A failed API call, model error, or timeout is an outage and goes to Observability.

Audit log

Each bot keeps an **Audit log** of who published, resolved, tagged, or changed something, and when. It is visible to Administrators and Security Admins, and it records every role change.

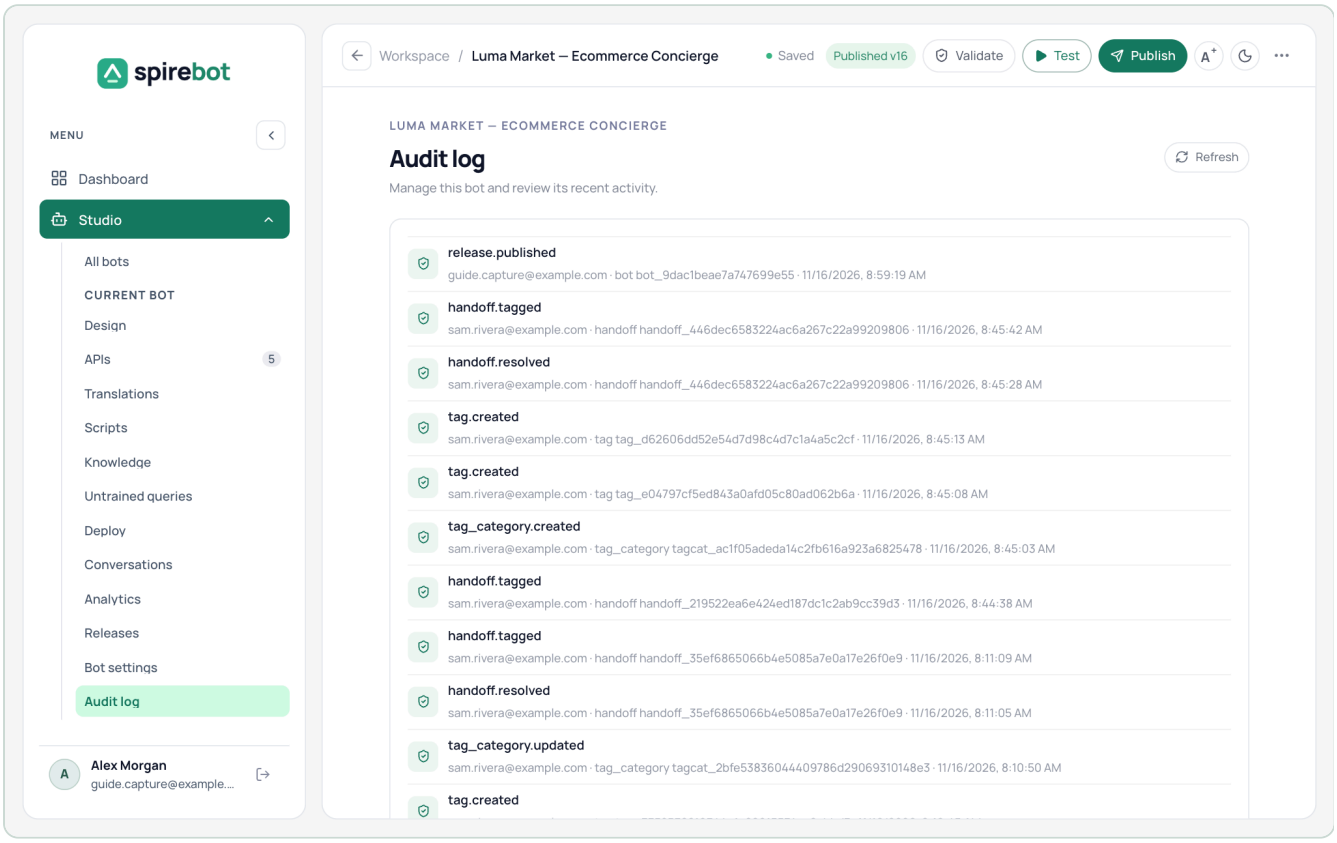


Figure 32. The audit log for a bot.

CHAPTER 11

Security and governance

Use the least data, access, and retention the bot needs to do its job.

Area	Practice
Secrets	Keep credentials in API connections. Never put tokens, passwords, cookies, or private keys in messages, variables, scripts, URLs, or step headers.
Access	Give each person only the roles their job needs, and combine roles rather than granting Administrator. Keep Administrator to a small group. Review roles and department assignments regularly.
Data	Collect as little personal data as possible, choose transcript storage and retention deliberately, and keep sensitive values out of traces and release notes.
Change control	Use validation, versioned releases, and the audit log. Have a second person review high-risk changes.

CHAPTER 12

Troubleshooting

Start with the visible symptom, then narrow the problem down.

Symptom	What to check
A menu item or bot section is missing	The menu follows your roles. Ask an Administrator or Security Admin to check the roles on your account.
Changes to an API are not saved, or you see "API configurations are read-only for your role"	Saved APIs need the API Integrator role. Ask an API Integrator to make the change, or ask for that role.
"Your role can view this bot but cannot change its design"	Your roles allow reading the bot only. Flow changes need the Bot Designer role.
The bot cannot be published	Open the validation report and work through each issue: one Start flow, one entry per flow, connected steps, valid branch targets, declared variables, complete translations, and a staffed department for Live Agent steps.
An API step follows the Failure branch	Inspect the error in the run trace. Check the workspace connection, the request path, the variables, the expected status, the response mapping, and whether the remote service is available.
Knowledge returns a guarded response	Confirm that a relevant, current source is ready. Test a specific question and improve the source wording. A guarded response is correct when the evidence is not sufficient.
A value is missing or has the wrong type	Check where the variable is declared and assigned. Review the input, API mapping, Set Variable, AI Router, or script output in the trace.
Untrained queries stays empty	Check that transcript storage is on in Bot settings. The list fills only from live conversations, not from the playground.
A trained question is still not answered	If it was added to an intent, publish the bot first. If it was added as an answer, add another answer that uses the customer's own wording.
Observability shows a worker as Down or Unknown	Unknown means the worker has not reported yet; Down means it has stopped. Ask your administrator to start the workers. Without the channel worker, WhatsApp messages are not delivered.
A close, fallback, or inactivity flow does not run	Confirm that the flow is selected in Bot settings and is part of the published release.

Symptom	What to check
An agent cannot claim a chat	The agent has reached their concurrent chat limit. Resolve a chat, or raise the limit under Users & agents.
The Agent inbox says “Agent access required”	Ask an Administrator or Security Admin to give you the Agent or Supervisor role in a department.
The draft works but the live bot does not	Check that the intended version was published and that the deployment is switched on. Then check the embed script or WhatsApp number and test from the real channel.

CHAPTER 13

Launch checklist

Complete this review for every meaningful release.

Design

- [ ] Goals and boundaries are documented.
- [ ] Every path ends, recovers, or hands over.
- [ ] Messages are concise and suit the channel.

Configuration

- [ ] Variables, languages, fallback, and conversation behavior are correct.
- [ ] Knowledge sources and connections are current.
- [ ] Secrets are stored only in API connections.

Quality

- [ ] Validation reports no issues.
- [ ] Happy, wrong-input, empty, failure, and handover paths pass.
- [ ] Traces and messages reveal no sensitive data.

Release

- [ ] The release note and owner are recorded.
- [ ] The deployment works in a test from the real channel.
- [ ] Observability shows every service and worker as healthy.
- [ ] Someone owns the Untrained queries review.