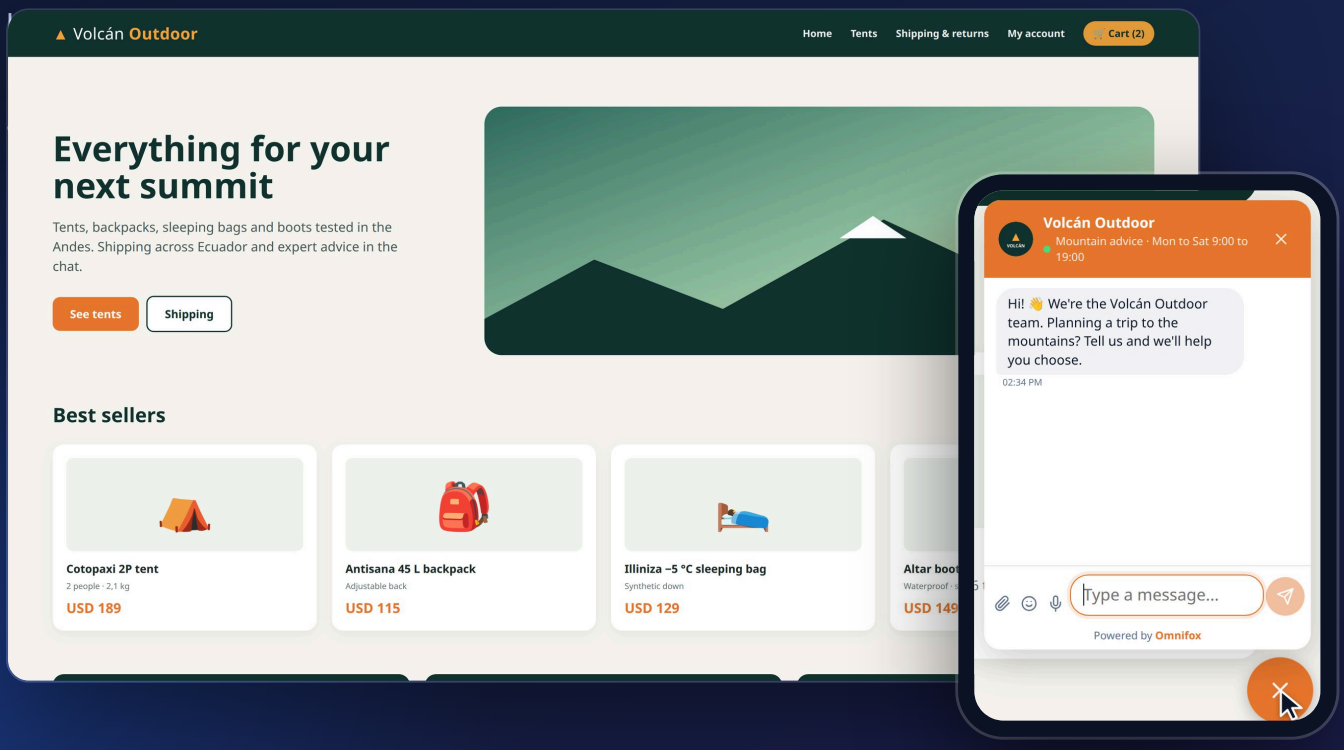
Omnifox

WEBCHAT · COMPLETE GUIDE

Your website chat

Create the channel, install it on your website, every appearance option, languages and AI translation, form and departments, business



CONTENTS

Table of contents

How to read this manual

The screenshots are real Omnifox screens, in English, and a sample website with the chat really installed: **Volcán Outdoor**, a fictional mountain gear store in Quito. Its AI agent is called **Vela**; its teams **Ventas** (sales) and **Soporte postventa** (after-sales support) are shown to visitors as **Buying advice** and **Shipping & warranty**; in the inbox, **Valeria Montes** and **Diego Paredes** answer. Customers and conversations are made up.

Button and field names are in **bold**, exactly as they appear on screen. Chapter by chapter, the manual follows the video "Webchat · Guía completa" on the Omnifox YouTube channel (in Spanish) and expands it with every field, every default value and every limit. Start with "Before you begin", on the next page.

01 What the Webchat is 4
02 Create the channel 6
03 Install it on your site 9
04 The channel in the list 12
05 Embed code and installations 13
06 Name, color, typography and position 14
07 Welcome message and avatar 16
08 Theme and branding 18
09 The chat button 19
10 Widget language 20
11 Two-way AI translation 22
12 Pre-chat form 23
13 Department selector 26
14 Business hours and out of hours 28
15 Visitor sound and notifications 29
16 Auto-open 30
17 Proactive bubble and advisor photo 33
18 Triggers: the first one step by step 34
19 All the conditions and AND/OR logic 37
20 All the actions, frequency and hiding 39
21 AI contextual greeting 40
22 The channel's AI agent 42
23 What the visitor can do 44
24 Handle it from the inbox 46
25 Resolve, survey and transcript 49
26 The returning visitor 50
27 On mobile 52
28 Secure user identification 53
29 JavaScript API and attributes 56
30 Co-browse 60
31 Good practices 60
32 Troubleshooting 61

✓

BEFORE YOU BEGIN

What you need and what your plan includes

Five minutes to have everything at hand before creating the channel.

What you need

- **An Omnifox account** and a user with permission to **manage channels**. Someone who can only view channels can open the widget editor, but sees a notice that they can only view this widget's settings and cannot save or see the identification secret key.
- **Access to your website's code** (or to whoever manages it): you paste one line of code on every page, right before the closing `</body>` tag. Any site works: hand-made, WordPress, Shopify, Wix, Webflow or your own store.
- **Teams created** in Settings → Teams, if you want the visitor to choose a department (chapter 13).
- **An AI agent**, if you want automatic replies (chapter 22). Creating and training it is explained in the complete guide to AI Agents.

What each plan includes

Feature	Available on	Detail
Webchat channel	All plans	Counts as one more channel within your plan's channel limit. You can have several widgets (one per site or brand).
Appearance, widget languages, pre-chat, departments, hours, triggers, identification and API	All plans	At no extra cost.
AI contextual greeting (chapter 21)	All plans	Uses the platform's AI engine; you don't need to create an agent.
AI agent that replies (chapter 22)	According to your plan's AI limits	Agents and their AI usage follow your plan.
Two-way AI translation (chapter 11)	From the **Crece** plan	With the plan there is a monthly quota of translated messages; as an add-on, each message is charged to your wallet (approx. USD 0.006).
Satisfaction survey in the chat (chapter 25)	According to your plan	Uses the workspace survey (Settings → Satisfaction survey).
Co-browse (chapter 30)	Add-on	See the visitor's screen and guide them, with their permission. It has its own complete guide.

> **NOTE**
>
> If your account is suspended for non-payment, the widget stops showing on your site until it is settled. It also doesn’t load for search engine bots or audit tools (Googlebot, Lighthouse): it doesn’t affect your SEO.

What the Webchat doesn't do

So there are no surprises, this **does not exist** today:

- An **allowed domains list**: the widget key works on any site where you paste it. That's why you should check **Detected installations** from time to time (chapter 5).
- A **consent step** (cookie or privacy notice) before chatting. If you need one, put it on your site.
- **Continuing the conversation on another device** with a link, or moving it from the chat to WhatsApp or email from the widget. The exception: an identified customer (chapter 28) is linked to the same contact record from any device.
- **Plugins** for WordPress, Shopify or Wix: you install it by pasting the code (chapter 3).
- **Different hours per department**: the hours belong to the whole widget.
- A **Webchat analytics screen** for your account: to measure, use the events the widget sends to Google Analytics 4 or Google Tag Manager (chapter 29).

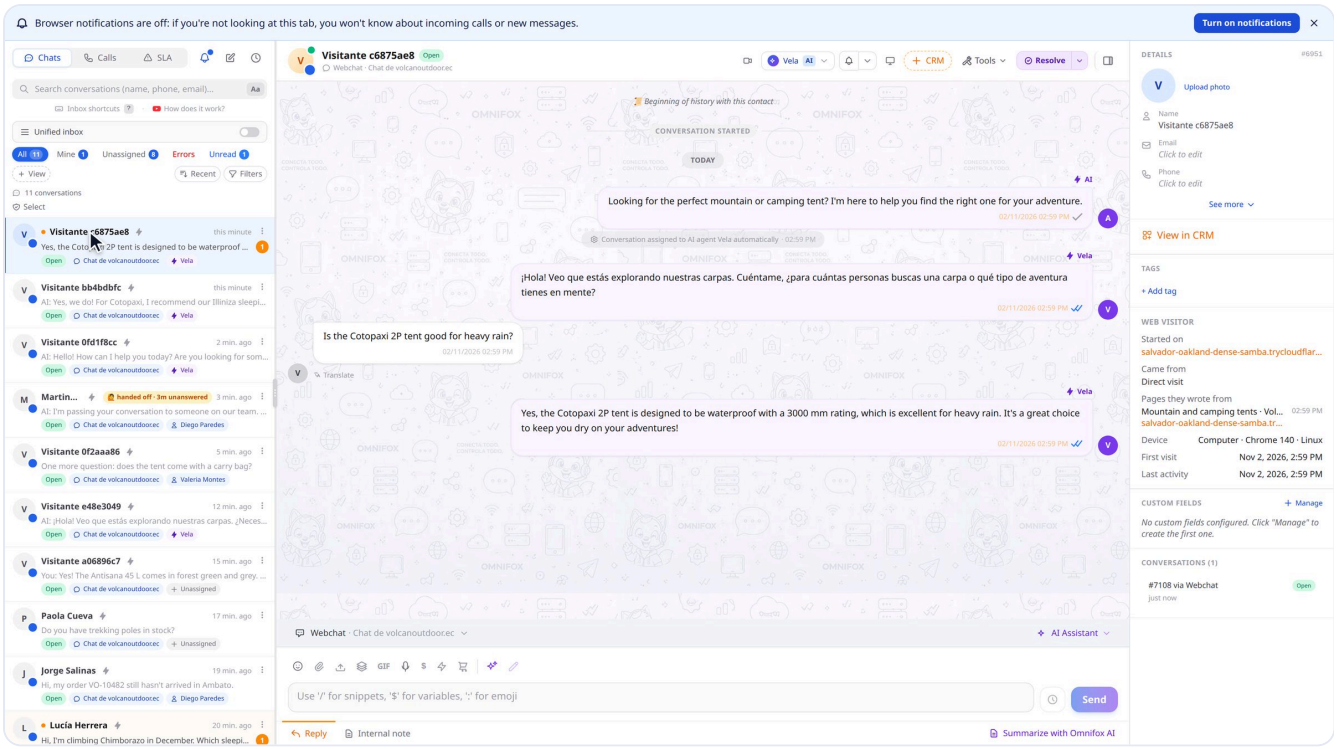
01

CHAPTER 1

What the Webchat is

A button on your website that brings every message to your Omnifox inbox.

The **Webchat** is a chat button that appears on your site. The visitor clicks, writes, and their message instantly reaches your **Omnifox inbox**, the same one where you handle WhatsApp, Instagram, Messenger, email and the other channels. There, someone from your team or an artificial intelligence agent answers it.



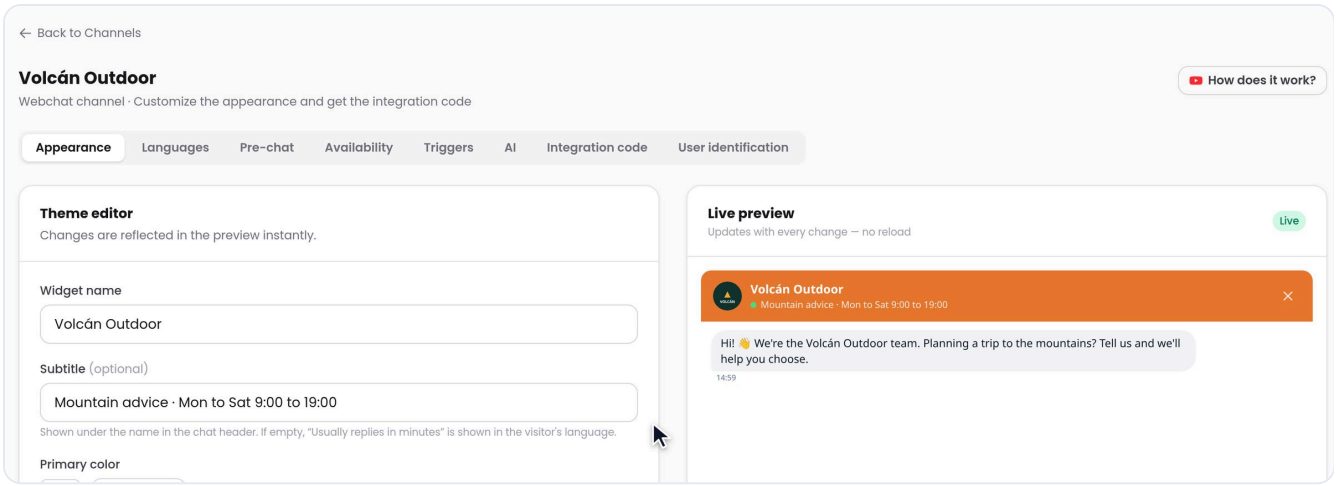
A visitor writes from the Volcán Outdoor store and the conversation reaches the Omnifox inbox, alongside the other channels.

What it can do

If you want…	Use	Chapter
The chat to look like your brand	Appearance: name, subtitle, color, typography, avatar, theme, button	6 to 9
To serve in Spanish, English or Portuguese	Widget language and Two-way AI translation	10 and 11
To know who they are before chatting	Pre-chat form	12
The visitor to choose who to talk to	Department selector	13
To reply differently when you are closed	Business hours and away message	14
To greet only on certain pages or moments	Auto-open and behavioral triggers	16 to 20
To reply instantly, 24 hours a day	AI greeting and the channel's AI agent	21 and 22
To recognize signed-in customers	Secure user identification	28
To open the chat from your own buttons and measure results	JavaScript API and GA4/GTM events	29

Where it is configured

Everything starts in **Settings → Channels**. After creating the channel, almost everything is set in the **widget editor** (the **Embed code** button on the channel card), which has eight tabs:



The widget editor: eight tabs with everything the chat can do.

Tab	What it contains	Chapters
Appearance	Name, subtitle, color, typography, position, welcome, avatar, theme, button and proactive bubble	6 to 9, 17
Languages	Widget language and AI translation	10 and 11
Pre-chat	Email transcript, form and departments	12, 13 and 25
Availability	Sound, notifications, hours and away message	14 and 15
Triggers	Auto-open and behavioral triggers	16, 18 to 20
AI	AI contextual greeting	21
Integration code	The line of code and Detected installations	5
User identification	Secret key, signature and the identify call	28 and 29

> **NOTE**
>
> The AI agent that answers messages is not chosen in the widget editor but on the channel's **Artificial Intelligence** screen (chapter 22).

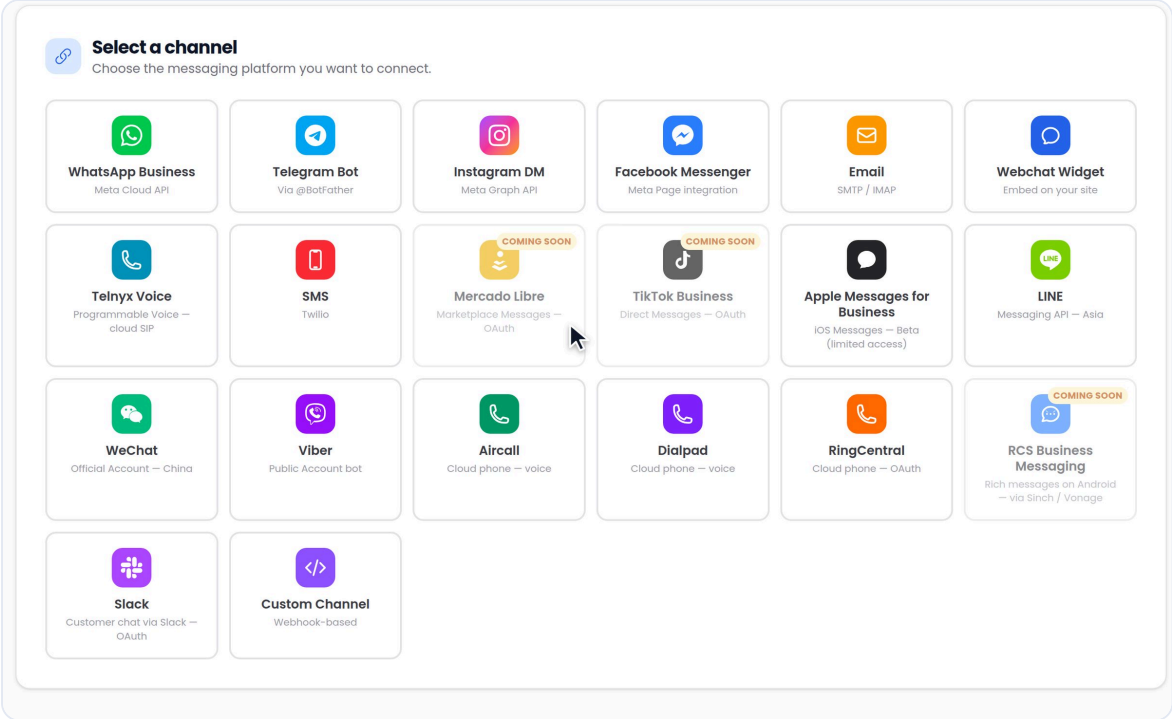
02

CHAPTER 2

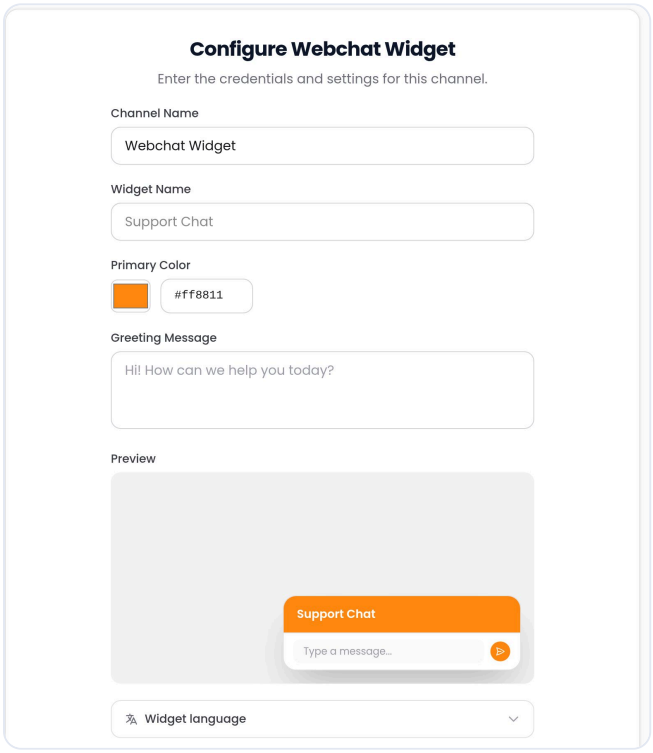
Create the channel

Settings → Channels → Connect new channel → Webchat widget.

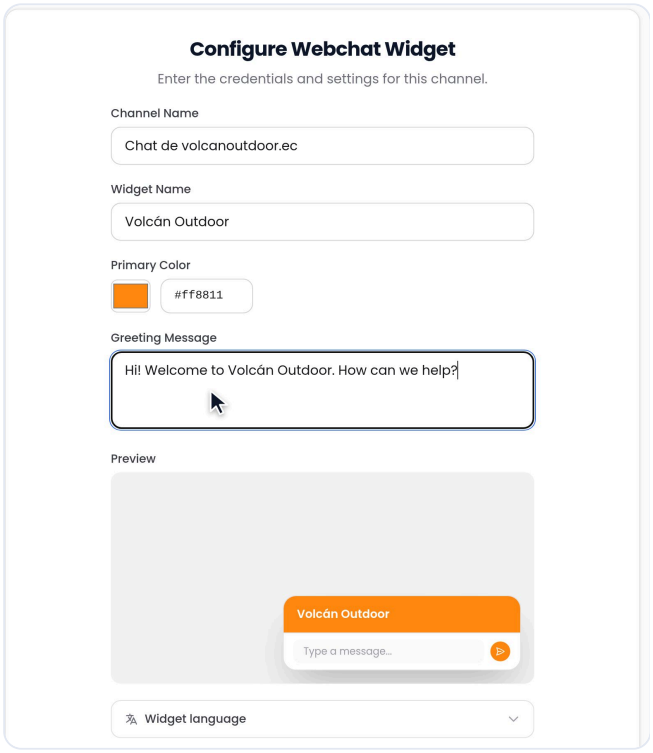
1. Go to **Settings** → **Channels** and click **Connect new channel**.
2. All the platforms you can connect appear. Choose the **Webchat widget** card ("Embed it on your website").
3. Fill in the **Configure** step (table below) and click **Save and get code**.
4. You reach the **Done** step, with your embed code ready to copy (chapter 3).



Settings → Channels → Connect new channel: the Webchat widget card.



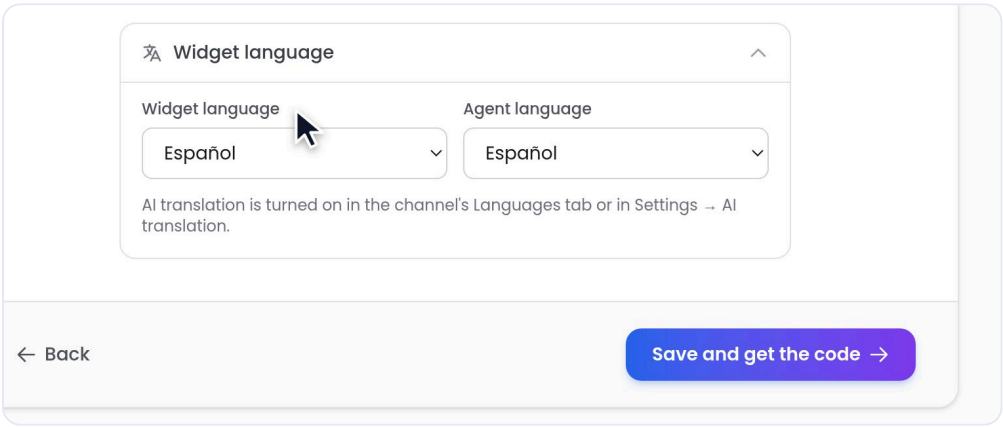
The wizard just opened: Channel name, Widget name, Primary color and Welcome message, with its preview.



The form filled in for Volcán Outdoor.

The wizard fields

Option	What it does	Default / limits
Channel name	How you'll see the channel inside Omnifox (channel list, inbox filters). Use, for example, the domain: "Chat de volcanoutdoor.ec".	Required · the visitor doesn't see it
Widget name	The chat header title, the one the visitor sees.	Required · max. 255 characters · e.g. "Support chat"
Primary color	Your brand color: button, header and visitor messages. Picker or hex code.	`#ff8811` (Omnifox orange)
Welcome message	The first thing the visitor reads when opening the chat (chapter 7).	Empty · e.g. "Hi! How can we help you today?"
Preview	A mock-up of the chat with the name, color and welcome. It's only a reference: the live view is in the editor (chapter 6).	—
Widget language (collapsible section)	Which language the visitor sees the chat texts in: Spanish, English or Portuguese (chapter 10).	Spanish
Agent language	Which language your team writes in. Only appears if your plan includes AI translation (chapter 11).	Spanish



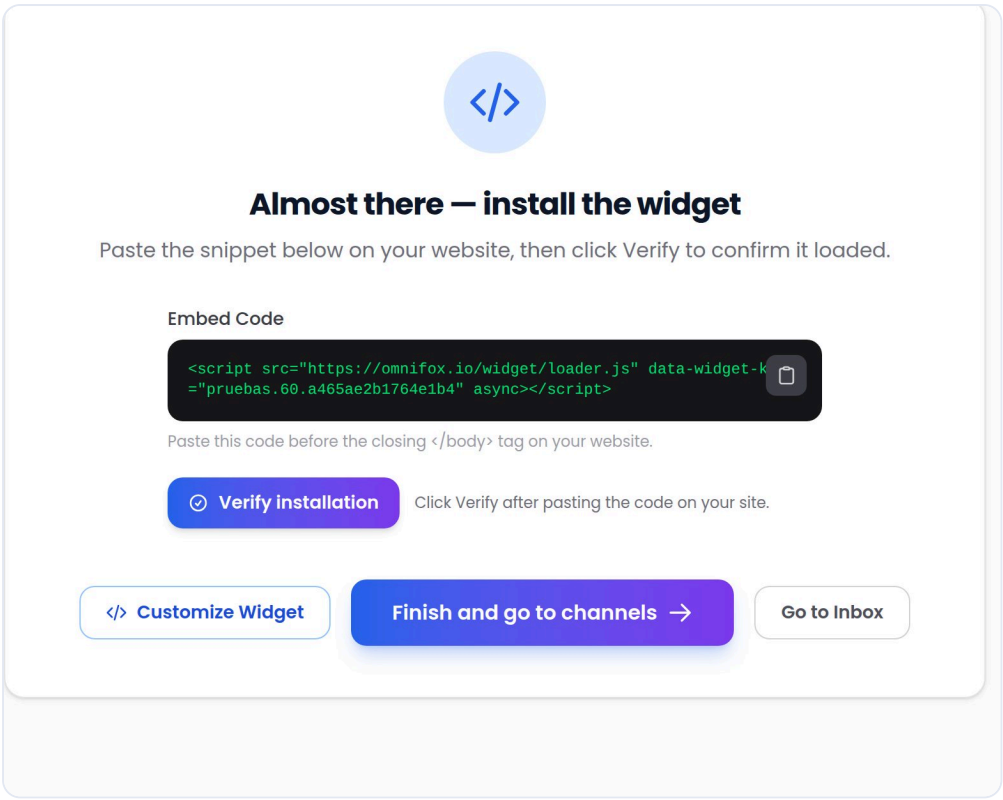
The wizard's Widget language section: the chat language and your team's.

> **NOTE**
> Automatic AI translation isn't turned on in the wizard: you activate it later, in the widget editor's **Languages** tab (chapter 11).

The Done step

When you save, the channel is created in pending status and the wizard shows **"Almost there — install the widget"**, with:

- the **Embed code**, with its copy button, and the reminder to paste it before the </body> tag on your website;
- the **Verify installation** button (chapter 3);
- three links: **Customize widget** (opens the widget editor), **Finish and go to channels** and **Go to the inbox**.



"Almost there — install the widget": the embed code, Verify installation and the three links at the end.

> **TIP**
>
> You can come back to this wizard at any time with the **Edit** button on the channel card (chapter 4). For everything else, use the widget editor.

CHAPTER 3

03 Install it on your site

A single line of code, on every page, before </body>.

The embed code is a single line. Copy it with the button and paste it on **every page** of your site, right before the closing `</body>` tag:

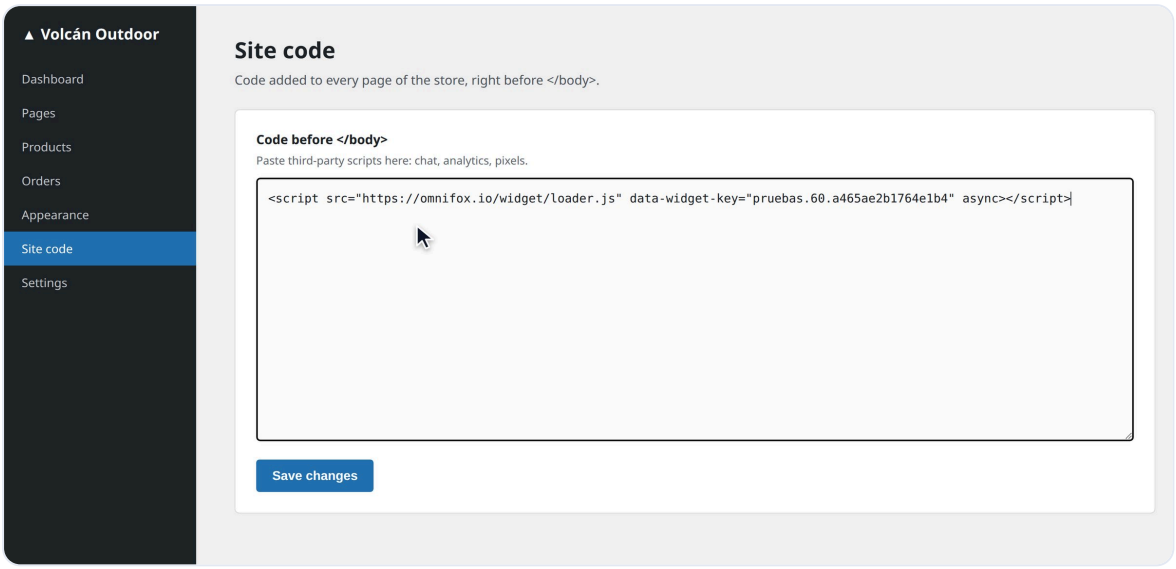
```
<script src="https://omnifox.io/widget/loader.js" data-widget-key="YOUR_WIDGET_KEY" async></script>
```

`YOUR_WIDGET_KEY` is your channel key (a combination of numbers and letters Omnifox fills in for you). It isn't secret, because it goes in your site's HTML, and it never changes.

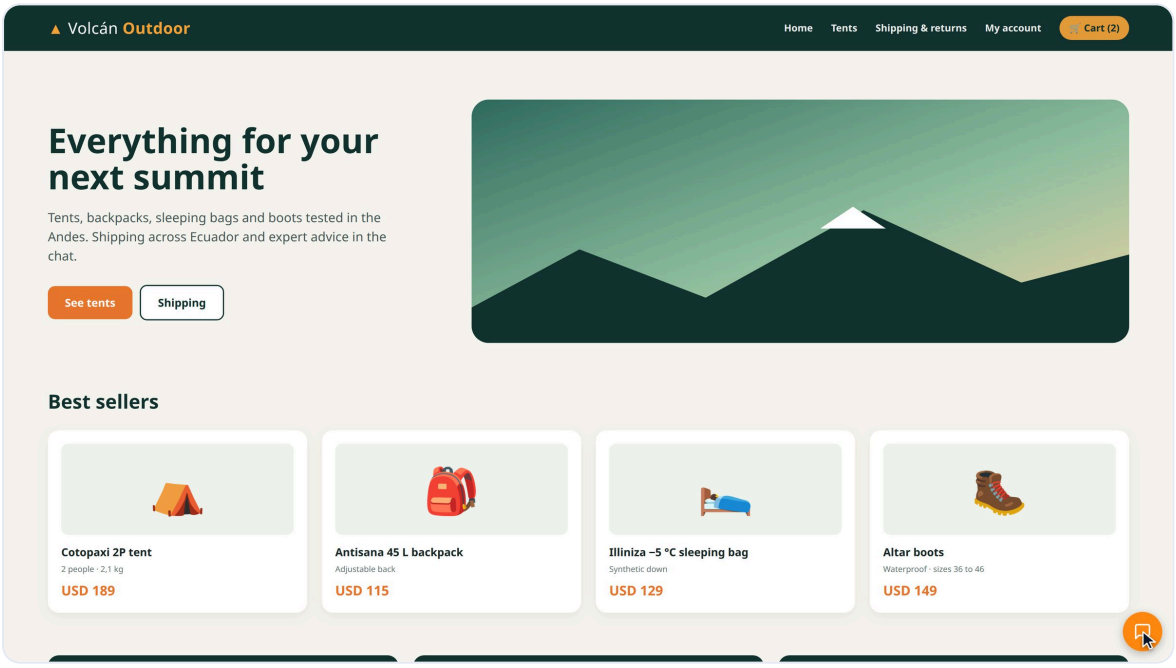
Step by step with the sample store

1. In the wizard's Done step (or in the **Integration code** tab), click the copy button.
2. Open your site's admin panel. The Volcán Outdoor store has a **Site code** section that adds code to every page, before `</body>`.

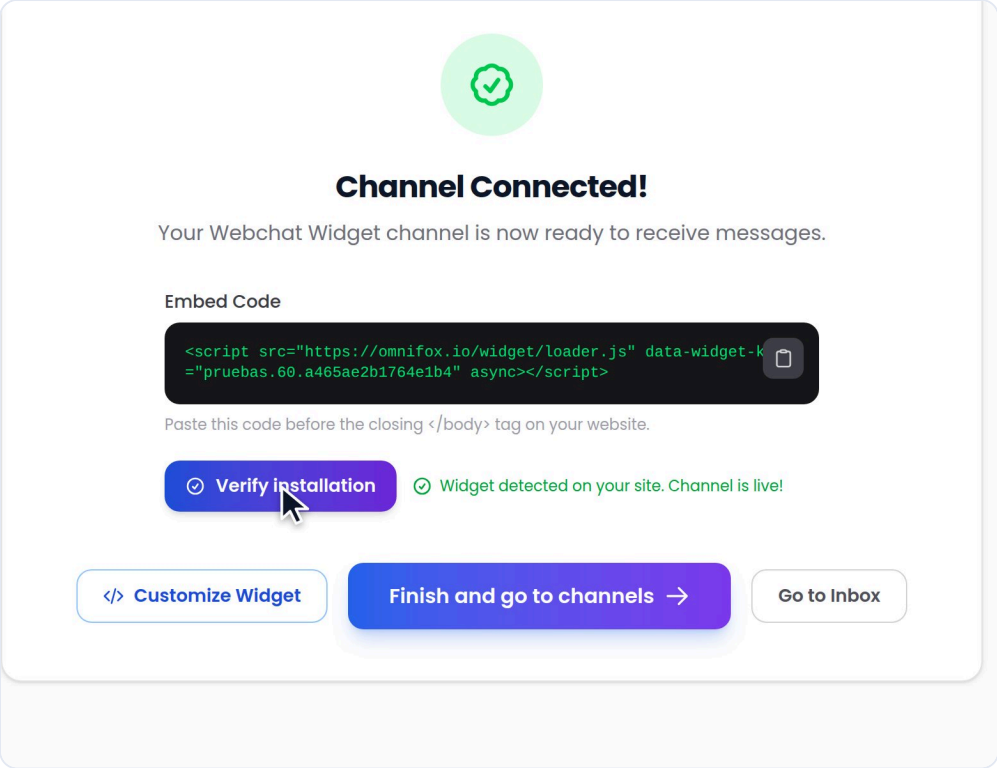
3. Paste the line and save.
4. Reload your site: the chat button appears at the bottom right.
5. Go back to Omnifox and click **Verify installation**.



The store panel: the “Site code” section with the line pasted before </body>.



The store reloaded: the chat button appears at the bottom right.



Verify installation: the widget was detected and the channel is active.

Verify installation checks that the widget has already loaded from some site after the channel was created. If all is well, you'll see that the widget was detected on *your domain* and the channel is active, and the title changes to **Channel connected!**. If it isn't detected yet, you'll see that the widget hasn't been detected: reload your site once with the code in place and try again.

Where to paste it, by platform

Platform	Where it goes
Hand-made site	In the shared template (footer or *layout*), right before `</body>`.
WordPress	In the theme footer (the child theme's `footer.php` file) or with a plugin that inserts code in the footer or header.
Shopify	Online store → Themes → Edit code → `theme.liquid`, before `</body>`.
Wix, Webflow, Squarespace and similar	The site's "custom code" or "embed code" option, applied to all pages and at the end of the body.
Google Tag Manager	A **Custom HTML** tag with the line, firing on all pages (All Pages).

```
<!-- Google Tag Manager → New tag → Custom HTML → Trigger: All Pages -->
<script src="https://omnifox.io/widget/loader.js" data-widget-key="YOUR_WIDGET_KEY" async></script>
```

> **NOTE**
> There is no official plugin for WordPress, Shopify or Wix: on all of them you install it by pasting the same line. If your site has a separate mobile version, paste it there too.

CAREFUL

Paste the line **only once** per page. If you change your site's theme or template, check that the code is still there: the Integration code tab tells you when it was last seen (chapter 5).

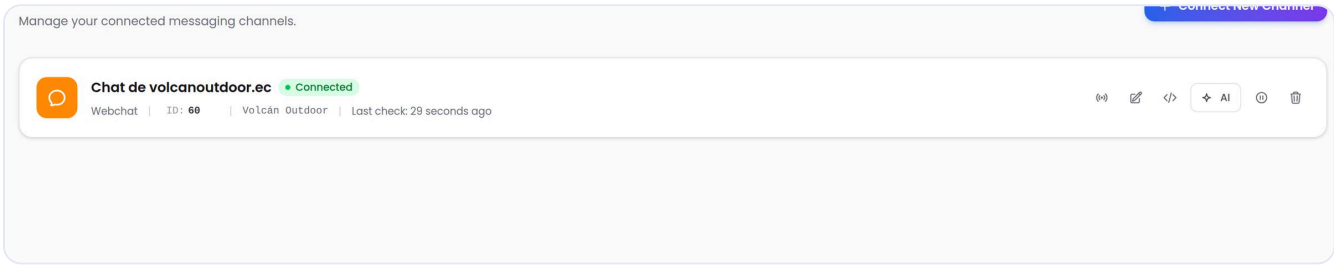
04

CHAPTER 4

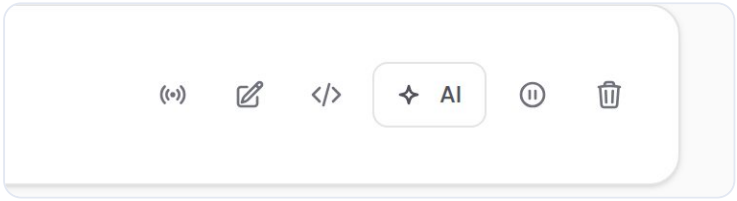
The channel in the list

The channel card: status, check, edit, code, AI, pause and delete.

In **Settings** → **Channels**, the Webchat has its card with the channel name, the type **Web chat**, its channel number, the widget name, the status (**Connected**) and the **Last check**.



The channel card in Settings → Channels: name, type, Connected status and last check.



The card buttons: health check, Edit, Embed code, AI, Pause and Delete.

Button	What it does
Health Check	Checks the channel status now and updates the last check.
Edit	Opens the same wizard as when creating it, in edit mode (channel name, widget name, color, welcome and language).
Embed code	Opens the **full widget editor**, with its eight tabs. It's where almost everything in this guide is done.
Artificial Intelligence	Choose the channel's AI agent and reply strategy (chapter 22). A green dot means the AI answers on this channel; an amber one, that an agent is assigned but the AI is off.
Pause channel / **Reactivate channel**	Turns the chat off for a while without losing anything: the widget stops showing and Omnifox stops monitoring the channel (no checks or alerts). When reactivated it comes back exactly as it was.
Delete channel	Deletes the channel and its settings. Asks for confirmation. Any code left on your site stops showing the chat.

CAREFUL

If you only want to turn the chat off for holidays or a change on your website, use **Pause**, not Delete: deleting loses the settings and, if you create it again, you'll have a new key to paste again.

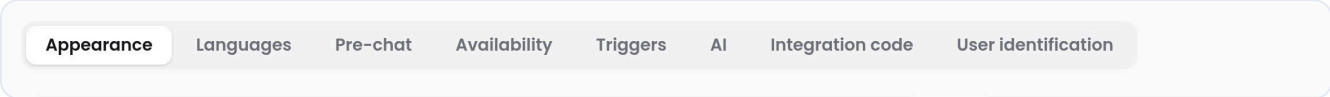
05

CHAPTER 5

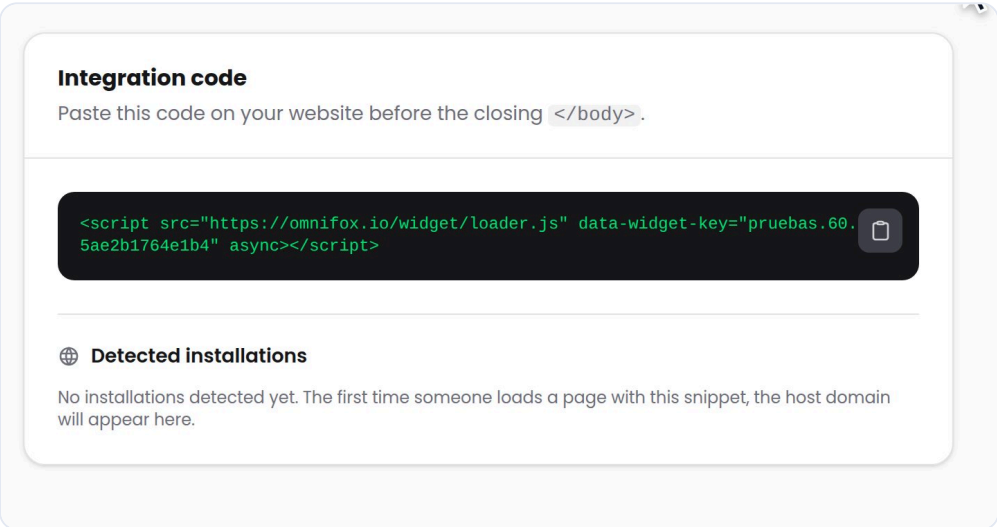
Embed code and installations

Your line of code always at hand, and the sites where it is installed.

The widget editor has eight tabs. The **Integration code** tab always keeps your line of code to copy it again (with the reminder to paste it before the closing </body> and its copy button), and below it shows **Detected installations**.



The eight tabs of the widget editor.



Integration code: the line to copy and Detected installations, with the most recent domain.

Detected installations

- Omnifox records every **domain** your widget was loaded from and **when it was last seen**.
- If you put it on several websites, you'll see them all; the most recent one carries the **Most recent** label.
- It updates at most once a minute, every time a visitor loads a page with the widget.
- If there are none yet, it says no installations have been detected yet.

TIP

Since there is no allowed domains list, Detected installations is how you know where your chat is. If a domain that isn't yours shows up, someone copied your code: contact support.

The code stays the same forever

You can change colors, texts, hours or triggers as often as you like **without touching your site again**: the widget reads its settings from Omnifox every time a page loads. To see a change, save the tab and reload your site.

> **CAREFUL**
>
> The code the wizard and the Integration code tab give you today only has `data-widget-key`. If you pasted an older code with `data-color`, `data-name` or `data-greeting`, those attributes **win** over what you save in Appearance: remove them so your changes show (chapter 29).

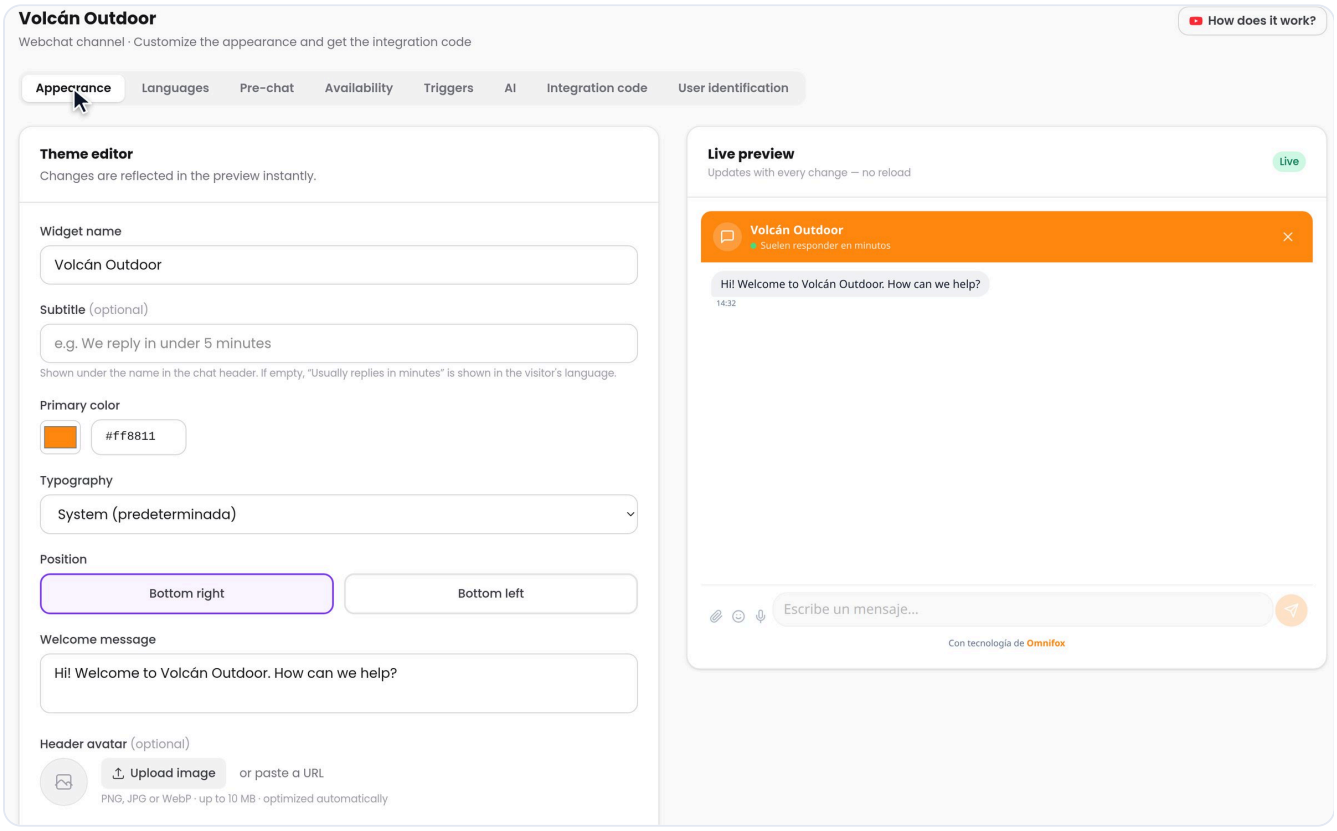
06

CHAPTER 6

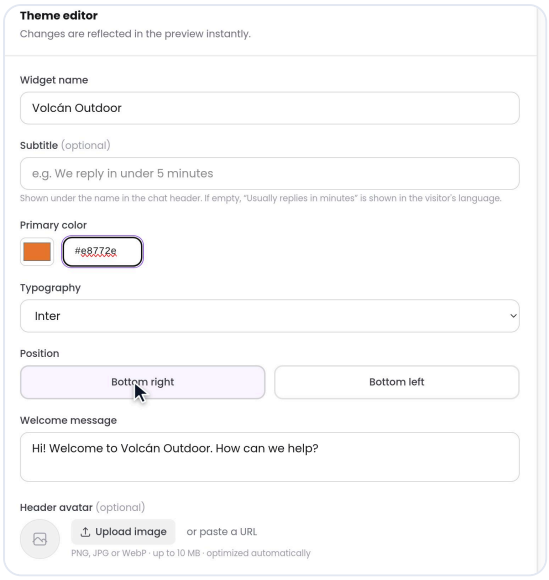
Name, color, typography and position

Appearance: the theme editor and the live preview.

In the **Appearance** tab you have, on the left, the **Theme editor** (changes show in the preview instantly) and on the right the **Live preview**, which changes with every setting without reloading.



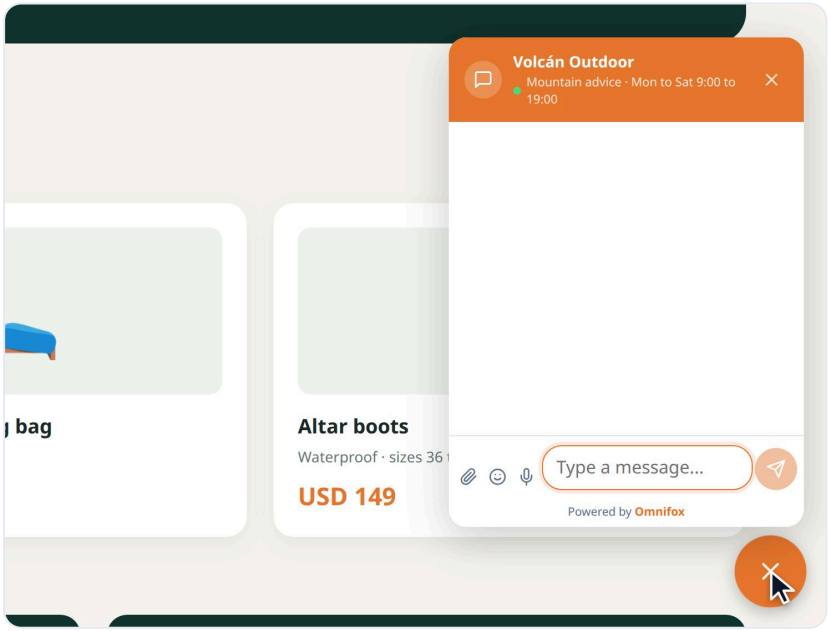
Appearance: the Theme editor on the left and the Live preview on the right.



Widget name, Subtitle, Primary color, Typography and Position.

Option	What it does	Default / limits
Widget name	The chat header title. It also renames the widget in the channel.	Required · max. 255
Subtitle (optional)	The small line under the name, in the header.	Empty = "We usually reply in minutes" (in the visitor's language) · max. 80
Primary color	Floating button, header and visitor messages. The text on top turns white or black by itself so it's always readable.	`#ff8811`
Typography	The chat font.	System (default) · Inter · Roboto · Open Sans
Position	Where the button sits and the chat opens.	Bottom right · Bottom left

1. Open Appearance and type the **Widget name** and, if you like, a **Subtitle** (in the example, "Mountain advice · Mon to Sat 9:00 to 19:00").
2. Choose the **Primary color** with the picker or type the code (in the example, `#e8772e`).
3. Choose **Typography** and **Position**.
4. Click **Save appearance** and reload your site.



The chat in the store, with Volcán Outdoor's color, typography and subtitle.

NOTE

The editor preview shows the header, colors, typography, avatar, message shape, theme mode, branding and welcome. The floating button, its text and the proactive bubble are shown in their own preview, further down, and the form or departments only on your site.

07

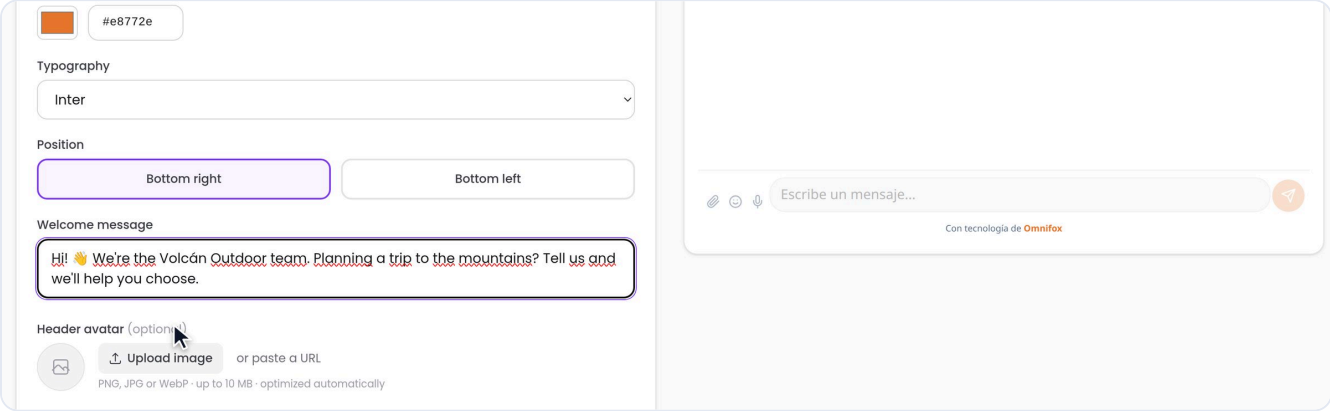
CHAPTER 7

Welcome message and avatar

The first thing the visitor reads, and the face of your chat.

Welcome message

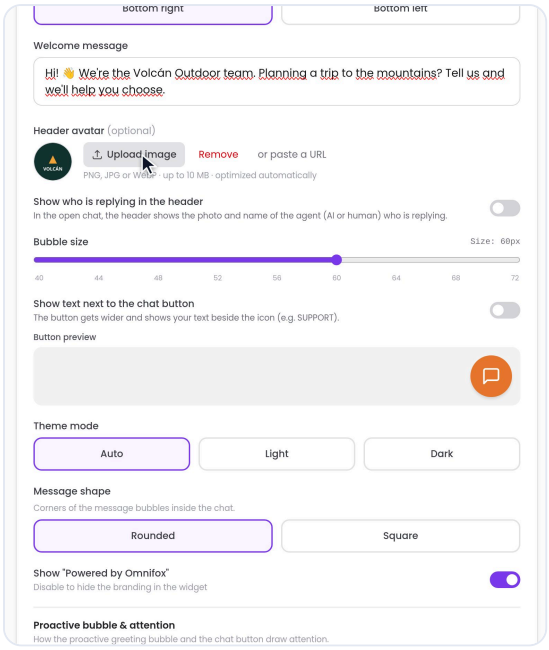
The **Welcome message** is the first thing the visitor sees when opening the chat, as if your team had written it. Maximum 500 characters.



The Welcome message; the preview shows it instantly.

- It's shown **only once per visitor**, in that browser: if they come back another day, it isn't repeated.
- It's shown **exactly as you write it**, untranslated: write it in your customers' language.
- If you turn on the **AI contextual greeting** (chapter 21), the AI replaces it; your text stays as a fallback.
- Until the visitor writes, the welcome lives only in their browser: it doesn't create a conversation in your inbox.

Header avatar and who is replying

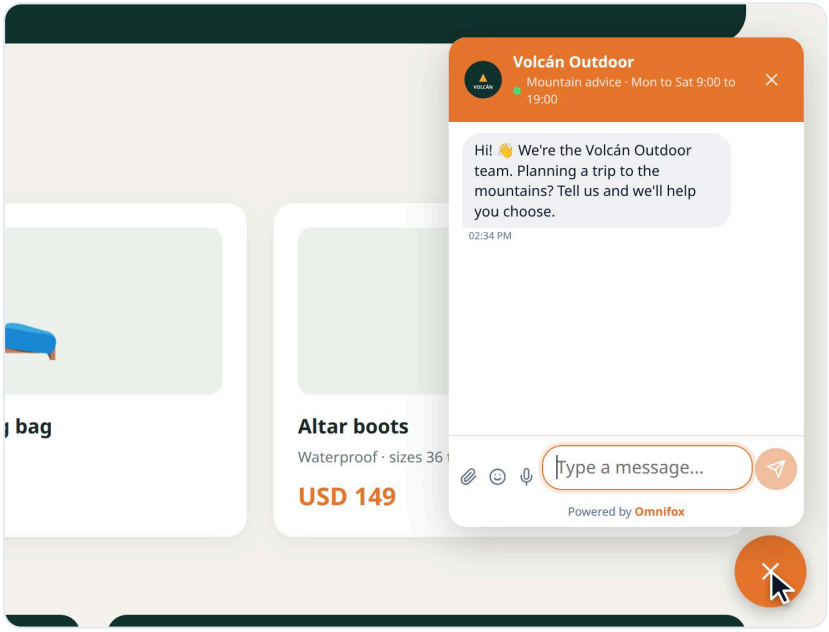


Header avatar uploaded, Show who is replying, Bubble size, Theme mode and Message shape.

Option	What it does	Default / limits
Header avatar (optional)	The round photo or logo in the header. **Upload image**, "or paste a URL", or **Remove**.	PNG, JPG or WebP · up to 10 MB · Omnifox resizes and optimizes it
Show who is replying in the header	With the chat open, when someone answers, the header switches to that person's or the AI agent's photo and name.	Off

TIP

Use a square logo with a background, at least 256 × 256 pixels: it's cropped into a circle.



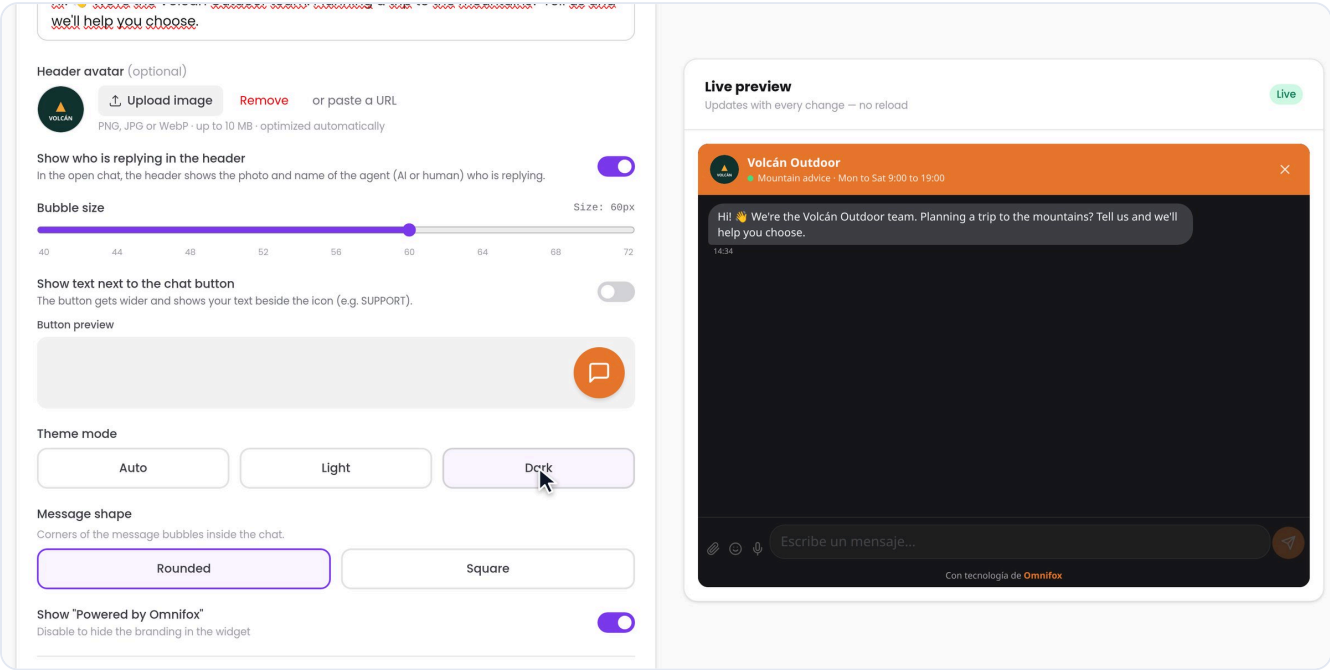
A new visitor opens the chat: the logo in the header and the welcome.

08

CHAPTER 8

Theme and branding

Light, dark or automatic; rounded or square messages; your brand or ours.



Dark theme mode in the preview.

Option	What it does	Default / limits
Theme mode	The colors of the chat panel. **Auto** follows whatever the visitor has chosen on their device.	Auto · Light · Dark (default: Auto)
Message shape	The corners of the message bubbles inside the chat. It doesn't change the floating button.	Rounded (default) · Square
Show "Powered by Omnifox"	The "Powered by Omnifox" footer inside the chat.	On
AI agent voice	The voice the AI agent uses to read its voice replies, with the **Try** button.	The workspace default voice

Save each tab

Each tab of the editor has **its own save button** (Save appearance, Save widget language, Save pre-chat settings, Save availability...). Saving one tab **doesn't save the others**.

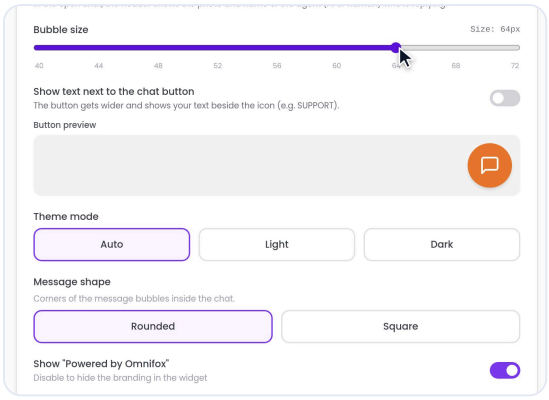
> **CAREFUL**
>
> If you change something in Appearance and go to another tab without saving, that change is lost. Always save before leaving the tab.

09

CHAPTER 9

The chat button

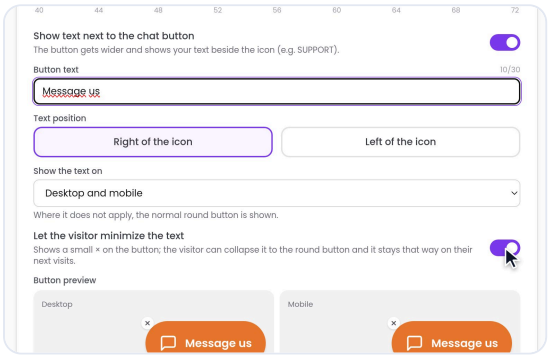
Size, text next to it, attention animation and the advisor’s photo as the icon.



Bubble size (40 to 72 px) and the button preview.

The floating button is the entrance to the chat. All its settings are in **Appearance**, below the avatar, with a **button preview** on desktop and on mobile.

Option	What it does	Default / limits
Bubble size	The diameter of the floating button.	60 px · from 40 to 72 px, in steps of 4

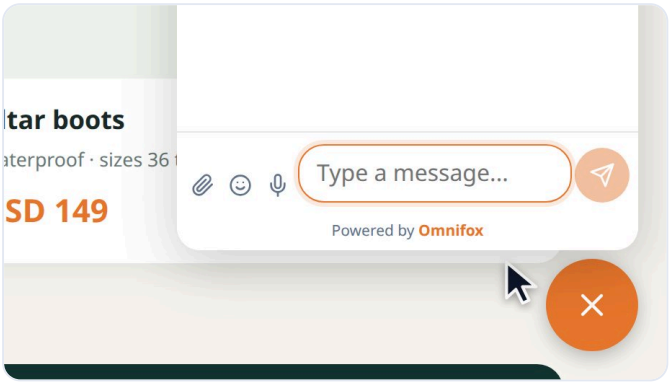


Show text next to the button: Button text, Text position, Show the text on and Let minimize.

Option	What it does	Default / limits
Show text next to the chat button	Turns the button into a “pill” with icon and text. If the text is empty, it stays round.	Off
└ **Button text**	The text, for example “Message us”.	Max. 30 characters
└ **Text position**	Which side of the icon it goes on.	Right of the icon · Left of the icon
└ **Show the text on**	Which screens show the pill; on the others, the normal round button.	Desktop and mobile · Desktop only · Mobile only (cut-off at 480 px wide)
└ **Let the visitor minimize the text**	Shows a small × on the pill: the visitor shrinks it to the round button and it’s remembered on their next visits. With the button already round, another × hides it for that page.	On

Animation and advisor photo

Option	What it does	Default / limits
Chat button attention animation	The button draws attention every so often.	None · Pulse · Bounce · Shake · Ring (heartbeat)
└ **Every … seconds**	How often the animation repeats.	12 s · from 3 to 120
Use the avatar as the chat button icon	Puts the advisor's photo (chapter 17) instead of the chat icon; if there isn't one, the header photo.	Off



The button with its text in the store.

TIP

An animation every 12 seconds or more draws attention without tiring. With "Shake" every 3 seconds, your visitor will leave before reading.

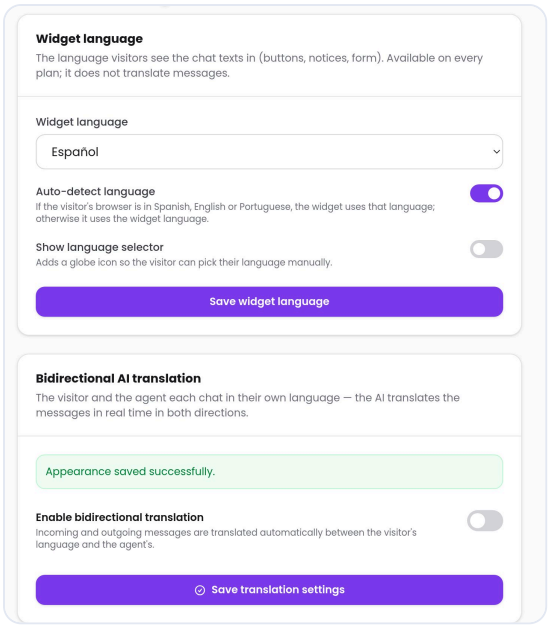
10

CHAPTER 10

Widget language

Spanish, English or Portuguese, on every plan.

In the **Languages** tab, the **Widget language** card decides which language the visitor sees the chat texts in: buttons, notices, form, out-of-hours message. It's available on **every plan**, has its own **Save widget language** button and **doesn't translate messages** (that's chapter 11).

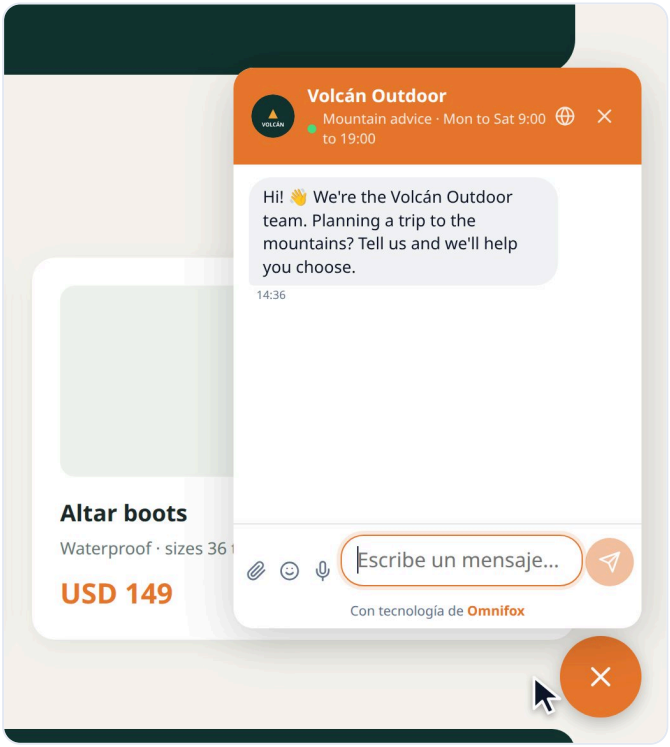


Languages tab: the Widget language card, on every plan, and below it Two-way AI translation.

Option	What it does	Default / limits
Widget language	The interface language when no other is detected.	Spanish · English · Portuguese
Detect language automatically	If the visitor's browser is in Spanish, English or Portuguese, the widget uses that language; otherwise, the Widget language.	On
Show language selector	A globe icon in the header so the visitor can choose their language by hand.	Off

Which language wins

1. **data-locale** → If the site page forces a language with the `data-locale` attribute, it always wins.
2. **Globe** → If the visitor chose a language in the header selector.
3. **Browser** → With automatic detection on, if the browser is in es, en or pt.
4. **Widget language** → The one you chose on this card.



A visitor whose browser is in Spanish sees the widget in Spanish, with the globe to change language.

Sites with several versions

If your site has pages in several languages (for example `/en/`), each version can force the widget language by adding `data-locale="en"` (or `"es"`, `"pt"`) to the line of code on those pages. To turn off browser detection on a page: `data-auto-detect="false"`.

> **CAREFUL**
>
> The language changes the widget texts, but **what you write** (welcome, subtitle, away message, form labels, trigger bubbles) is shown as is. Write it in your customers' language. Proactive greeting bubbles do have a version in all three languages (chapter 16).

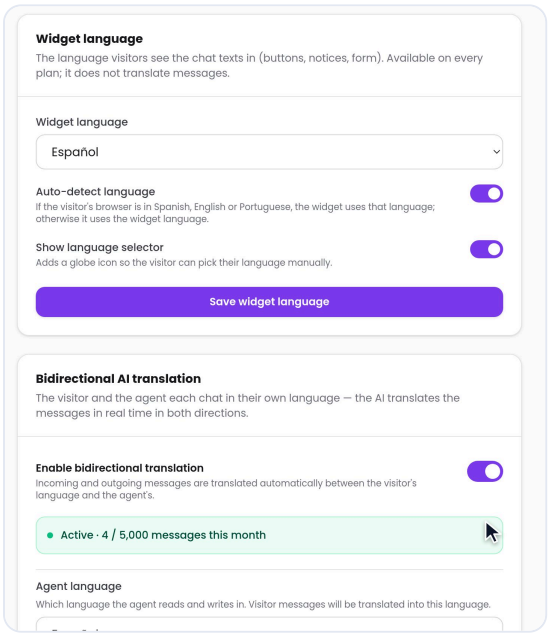
11

CHAPTER 11

Two-way AI translation

Visitor and agent each talk in their own language. From the Crece plan.

Two-way AI translation lets the visitor and your agent each talk in their own language: the AI translates incoming and outgoing messages in real time. It's available **from the Crece plan** (or as an add-on). If your plan doesn't include it, you'll see that the feature isn't available on your plan, with a button to see plans and upgrade.



Translation turned on: the month's usage, the agent language and the visitor language.

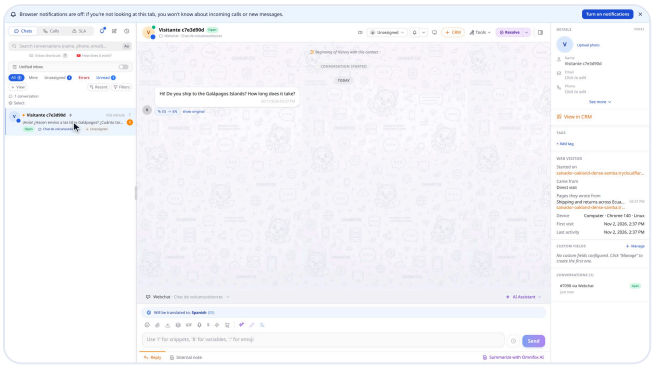
Option	What it does	Default / limits
Enable two-way translation	Translates between the visitor's language and the agent's. When on, it shows the usage: active · X / Y messages this month (or unlimited, fair use), or with the add-on the cost per message (~$0.006) and your balance, with **Top up wallet**.	Off
Agent language	The language your team reads and writes in. The visitor's messages are translated into it.	Spanish
Visitor language (default)	The one assumed when the visitor's can't be detected.	Same as the agent
Show "Translated by AI" notice	A "Translated by AI" footer inside the chat, above "Powered by Omnifox".	On

Save with **Save translation**. The fields only appear with translation turned on.

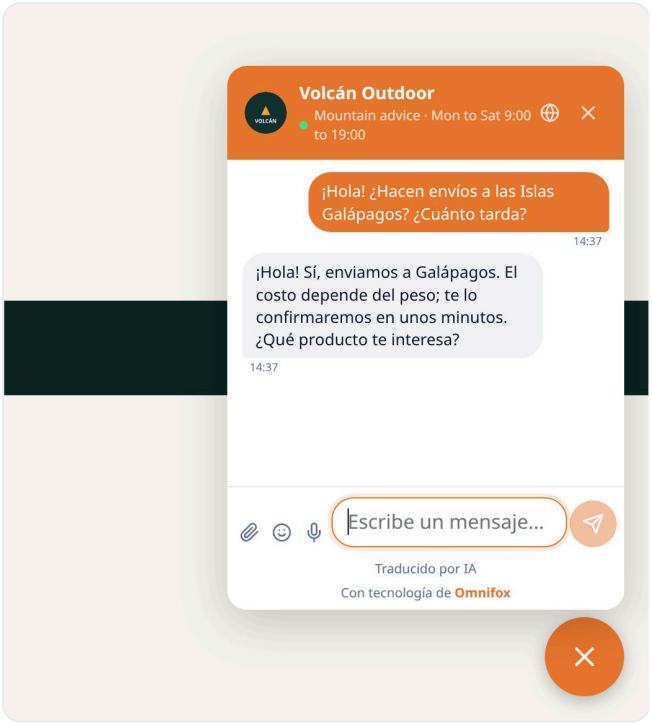
What it looks like

1. A visitor writes in Spanish: "¡Hola! ¿Hacen envíos a las Islas Galápagos? ¿Cuánto tarda?".

2. In the inbox, Valeria reads it in English.
3. Valeria replies in English: “Hi! Yes, we ship to the Galápagos...”.
4. The visitor gets the reply in Spanish, with the “Translated by AI” notice.



In the inbox, Valeria reads the message in English.



The visitor gets the reply translated into her language, with the “Translated by AI” notice.

TIP

Each translated message counts toward your plan’s monthly quota (or is charged to the wallet if it’s an add-on). If your customers speak your language, don’t turn it on: the Widget language from chapter 10 is free.

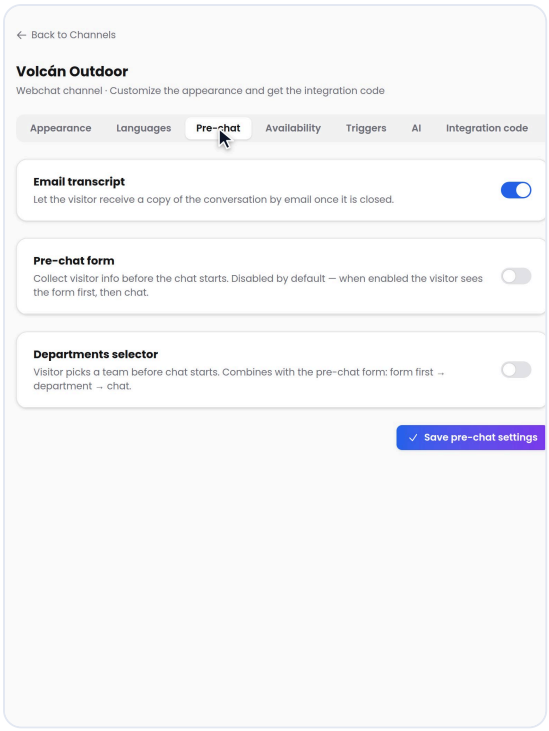
12

CHAPTER 12

Pre-chat form

Ask for just the details you need before chatting.

In the **Pre-chat** tab, the **Pre-chat form** asks for details before starting. It’s **off** by default; when you turn it on, you add up to **10 fields** with **Add field**. The widget introduces it with a short line asking the visitor to tell you a bit about themselves, and the **Start chat** button.



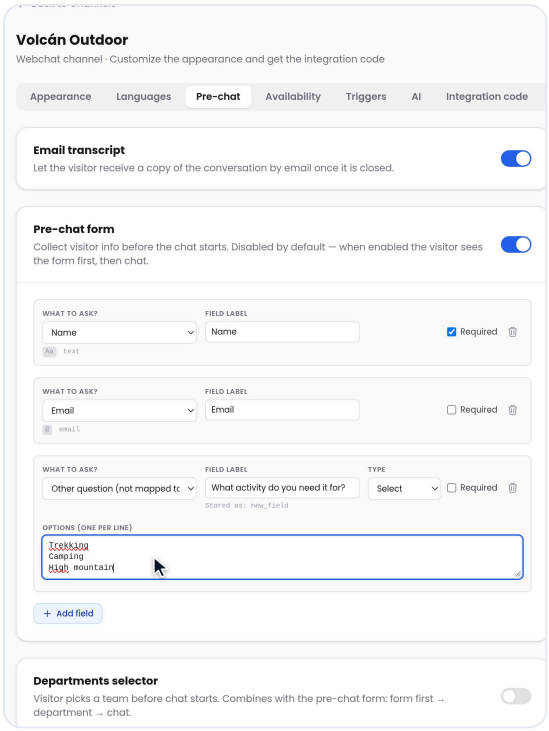
Pre-chat tab: email transcript, pre-chat form and departments selector.

The tab has three blocks, saved together with **Save pre-chat settings**:

- **Email transcript** (chapter 25).
- **Pre-chat form** (this chapter).
- **Departments selector** (chapter 13).

Each field

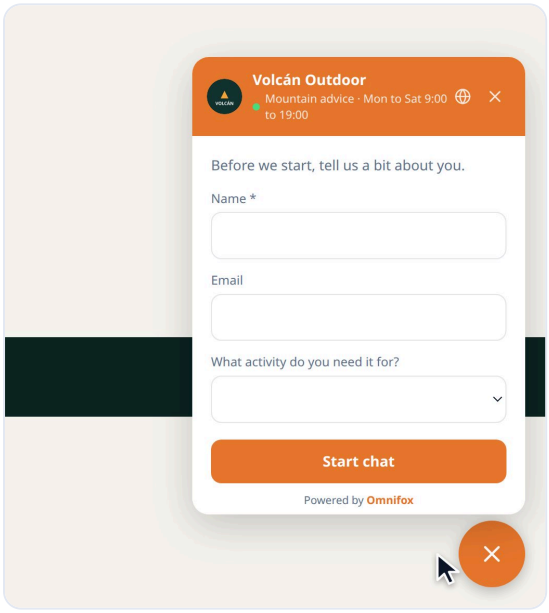
Option	What it does	Default / limits
What to ask?	**Standard contact fields**: Name, Email or Phone. **Custom contact fields**: the ones in your workspace. **Other question**: a free question.	—
Field label	What the visitor sees.	Max. 128 characters
Saved as	(Free questions only.) The internal name of the answer; derived from the label.	Max. 64
Type	(Free questions only.) What kind of answer.	Text · Text area · Number · Email · Phone · Select
Options (one per line)	(Select type only.) The options in the list.	One per line
Required	The visitor can't start without filling it in.	Unchecked
Delete row	Removes the field.	—



Three fields: Name (required), Email and a free question of type Select.

In the example: **Name** (required), **Email** and the free question "What activity do you need it for?", of type **Select**, with the options Trekking, Camping and High mountain.

Field type	Where the answer goes
Standard (Name, Email, Phone)	In the **contact record**.
Custom	In the record's **custom field**.
Other question	In the **conversation**: the **Pre-chat** section of the contact panel, in the inbox.



The form as the visitor sees it, with the Start chat button.

This is how the visitor sees it: first the form and, after tapping **Start chat**, the chat. The conversation reaches the inbox with the name and email already on the record.

NOTE

With the form **off**, the widget still works: the conversation starts with the first message and the contact shows as **Visitor** followed by a code (for example "Visitor 3d0e1954") until you know their name.

TIP

Every extra field is one visitor less. Ask for the name and one way to contact them; the rest you ask while chatting.

CAREFUL

The form only shows when the visitor **doesn't have** an open conversation. If they already chatted, they aren't asked again: to test your changes, use an incognito window.

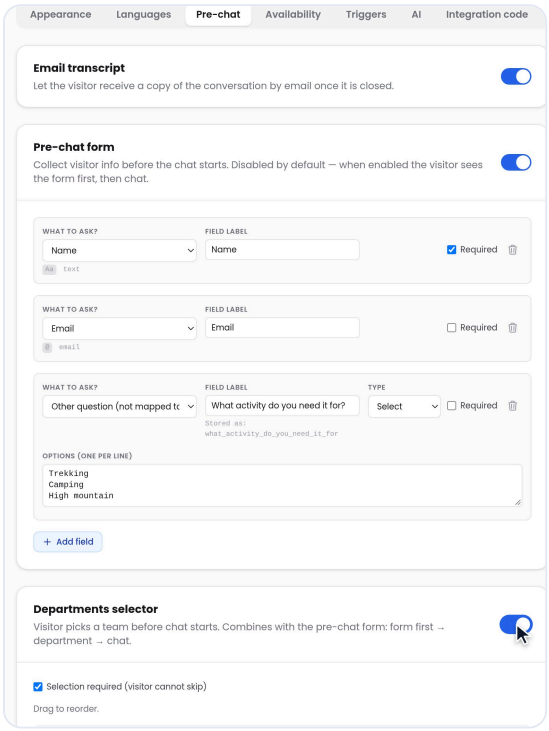
13

CHAPTER 13

Department selector

Let the visitor choose which team to talk to.

The **Departments selector**, in the **Pre-chat** tab, lets the visitor choose which team they want to talk to. The departments are **your teams** in the workspace: you tick which ones to show and how.



The workspace teams as departments: checkbox, display name, color and icon.

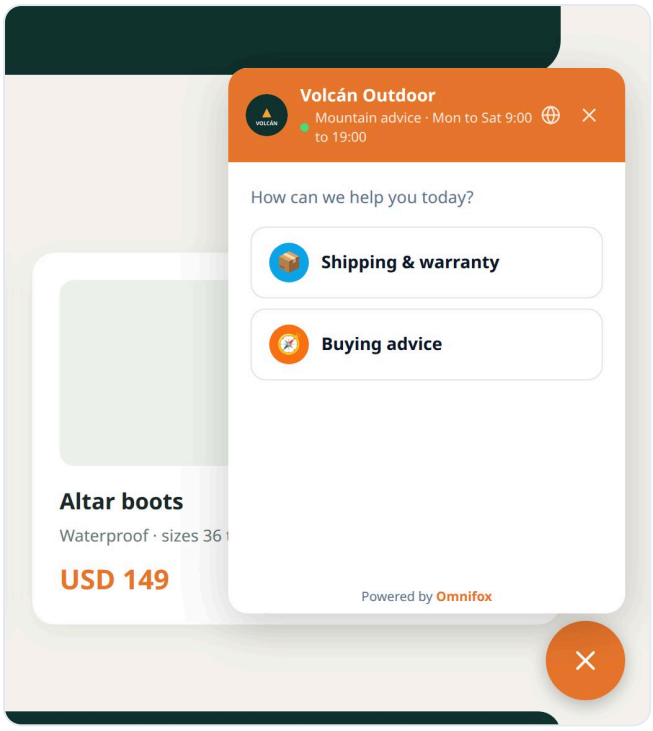
Option	What it does	Default / limits
Departments selector	Turns on the “Which team do you want to talk to?” step.	Off
Selection required	The visitor cannot skip it. When off, **Continue without choosing** appears.	On
Each team’s checkbox	Whether it shows in the widget.	—
Display name	How the visitor sees it (Ventas → “Buying advice”).	Empty = team name · max. 64
Color	The color of its button.	—
Icon	An emoji or short text (🧭, 📦).	Max. 64
Drag	Reorders the departments; dropping saves by itself.	—

The visitor’s order

Pre-chat form → If active: the details first.

Department → Then, the team.

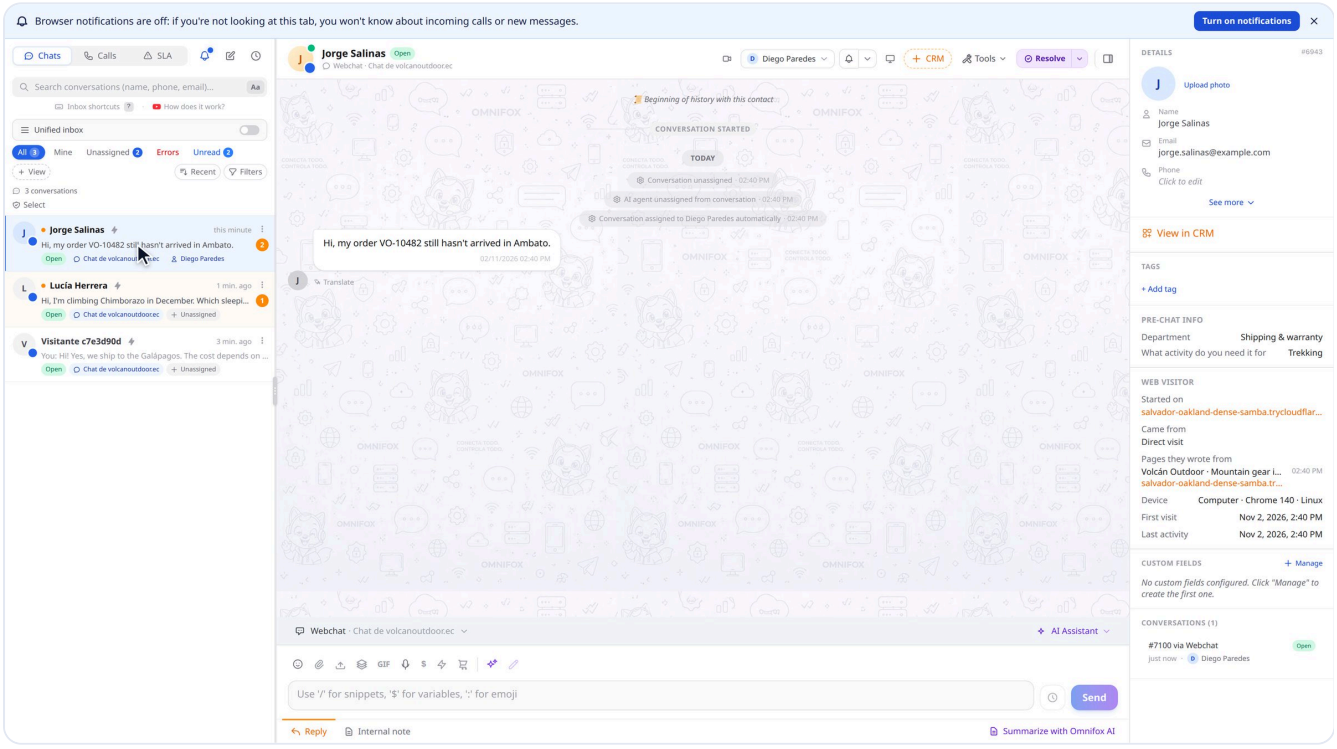
Chat → And finally, the conversation.



"Which team do you want to talk to?": the departments in the widget.

The visitor chooses **Shipping & warranty** and writes "Hi, my order VO-10482 still hasn't arrived in Ambato.". The conversation comes in **assigned to the Soporte postventa team** and is distributed according to the team's strategy (round robin, least busy...). Here it went to Diego.

Choosing a department doesn't turn off the channel's AI agent: if you have one with the Always strategy, it keeps answering until a person takes the conversation (chapter 22).



The conversation comes in assigned to the Soporte postventa team and distribution gives it to Diego.

> **NOTE**
>
> If you don't have teams yet, the selector says so: create a team first in Settings → Teams, with its members and how it distributes work, and come back.

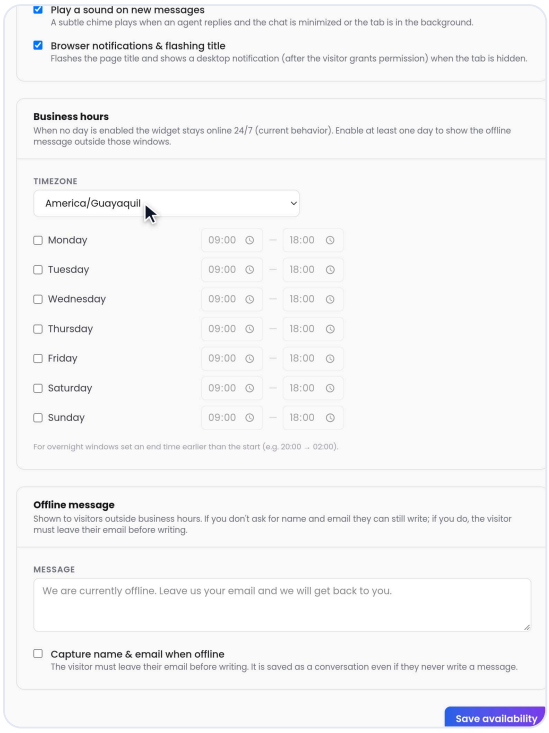
14

CHAPTER 14

Business hours and out of hours

Your hours, your away message, and no contact lost.

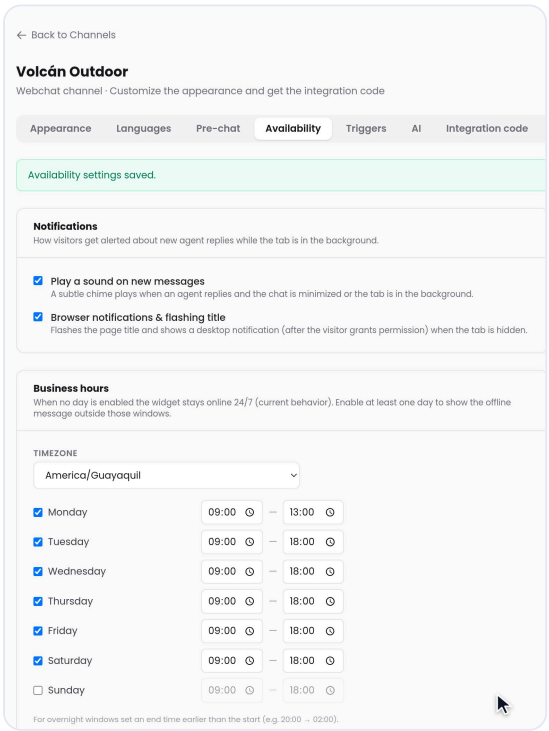
In the **Availability** tab you set when you're open. Save with **Save availability**.



Availability → Hours: time zone and, per day, whether you're open and at what times.

Option	What it does	Default / limits
Time zone	The zone your hours are calculated in.	Your account's · full list of zones (America/Guayaquil…)
Monday … Sunday	For each day: an "open" checkbox and start and end time.	All off · 09:00 to 18:00
Night hours	If the end is earlier than the start (22:00 to 06:00), the slot crosses midnight.	—

> **NOTE**
>
> If you don't turn on **any day**, the chat always shows as available, 24 hours a day.

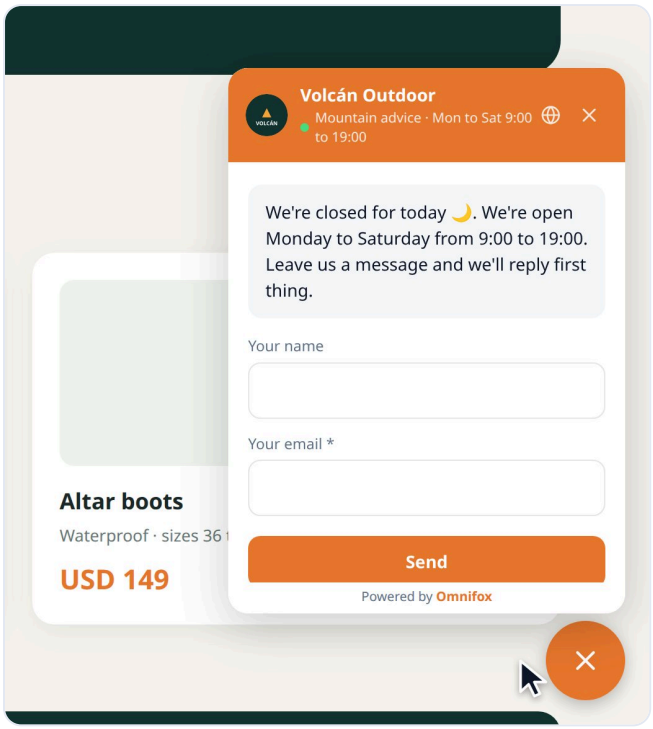


Away message and Capture name & email when offline.

Option	What it does	Default / limits
Away message	What the visitor sees outside business hours, as a message from your team.	Empty = a standard "we're out of hours right now" text · max. 1000
Capture name & email when offline	The visitor must leave their name and email before writing. It's saved as a conversation even if they don't write any message.	Off

Without the capture, outside business hours the visitor can still write: the message only sets expectations.

What the visitor sees



Out of hours: the away message and the name and email form.

In the "we're closed for today" example: the chat shows the away message and asks for name and email. Paola leaves them, writes "Do you have trekking poles in stock?", and the conversation still reaches the inbox to be answered the next day.

> **TIP**
> An honest away message says when you're back ("We're open Monday to Saturday from 9:00 to 19:00") and what will happen ("we'll reply first thing").

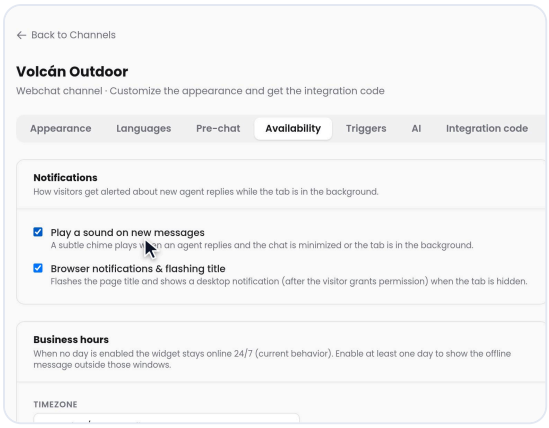
> **NOTE**
> The widget has a single schedule for the whole chat; there are no hours per department. If your AI agent has its own schedule, that one rules when the AI answers (chapter 22).

CHAPTER 15

15 Visitor sound and notifications

So your reply doesn't go unnoticed.

At the top of the same **Availability** tab, the **Notifications** block decides how the visitor finds out that you replied:

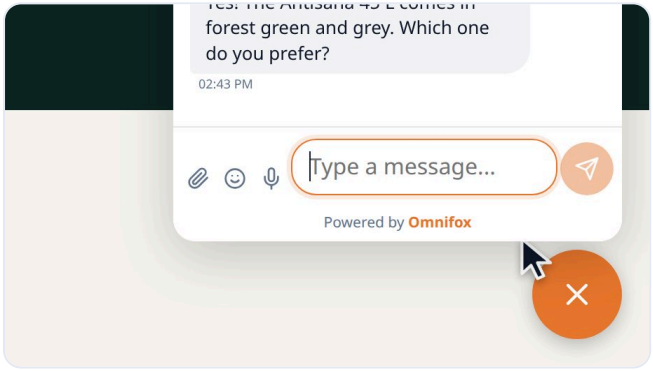


Visitor notifications: sound, blinking title and browser notification.

Option	What it does	Default / limits
Play a sound on new messages	Sounds when your team replies and the chat is minimized or the tab is in the background (at most one every 2 seconds).	On
Browser notifications and blinking title	The tab title blinks and, if the visitor allows it, a desktop notification appears.	On

Always on

- **Unread counter**: the chat button shows a red circle with the unread messages.
- **Reopening with the new message**: if the visitor closed the chat and your agent replies, the chat reopens by itself with the new message, so they don't miss it (only for new messages, not those in the history).



With the chat minimized, the button shows the unread messages counter.

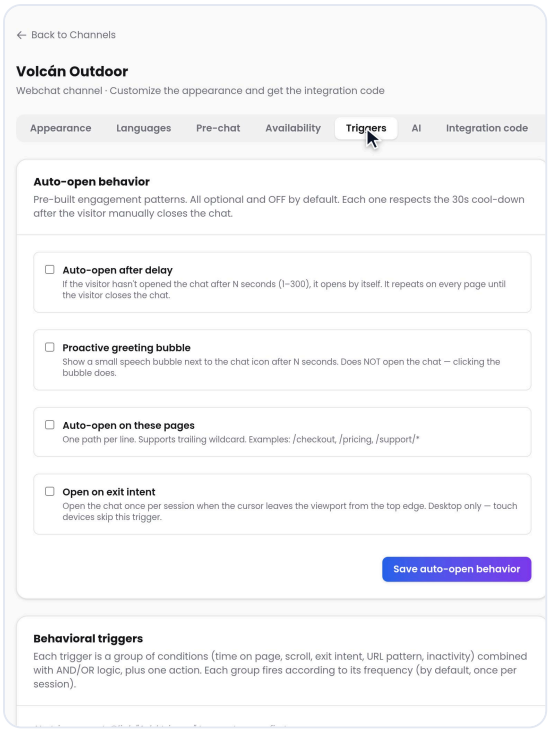
In the example, with the chat minimized, Valeria replies "Yes! The Antisana 45 L comes in forest green and grey" and the counter with the unread message appears in the store.

> **NOTE**
> The browser only lets you ask for notification permission after the visitor interacts with the page. If they declined, they still get the sound, the blinking title and the counter.

16 CHAPTER 16 Auto-open

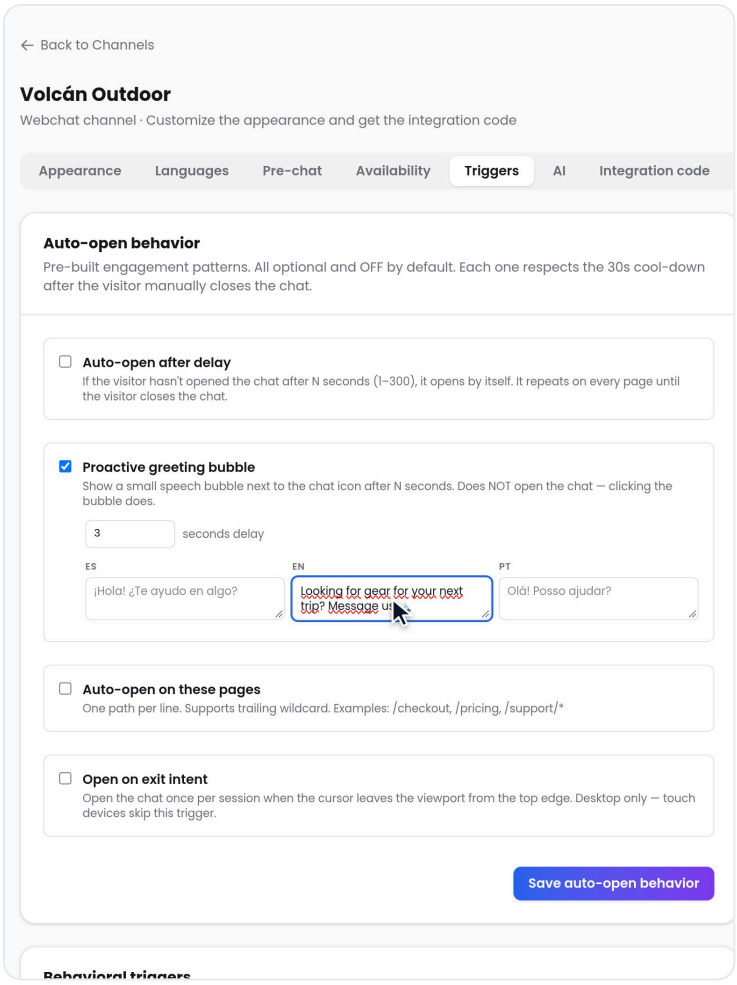
Four ready-made behaviors to open the chat at the right moment.

The **Triggers** tab starts with the **Auto-open behavior** block: four ready-made behaviors, **all off** by default. They're saved with their own save button.

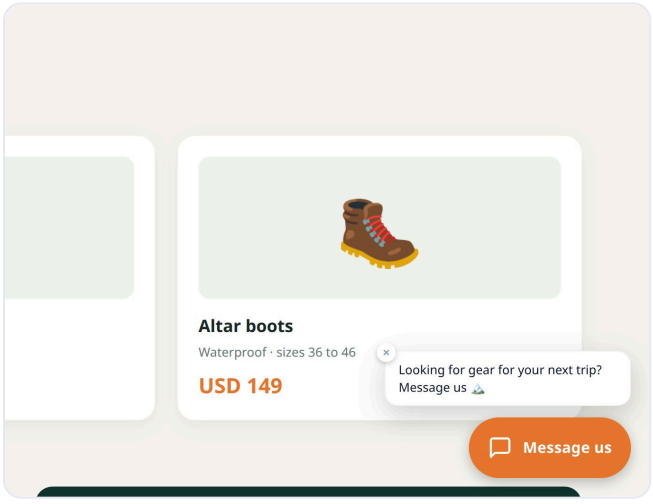


Triggers → Auto-open behavior: the four behaviors, off by default.

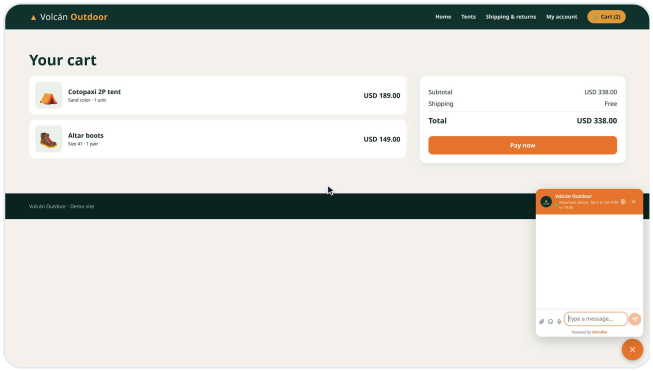
Option	What it does	Default / limits
Auto-open after a delay	If the visitor hasn't opened the chat after N seconds, it opens by itself. It repeats on every page until the visitor closes the chat.	Off · when ticked, 30 s · from 1 to 300
Proactive greeting bubble	A message next to the button, without opening the chat, with its delay and its text in Spanish, English and Portuguese. A click on the bubble opens the chat.	Off · 15 s · 0 = instantly · max. 200 characters per language
Auto-open on these pages	Opens the chat on the paths you write, one per line (`/checkout`). A trailing wildcard is allowed: `/support/*` . It's compared with the path, without domain or parameters.	Empty
Open on exit intent	Opens the chat when the cursor moves up, about to close the tab, with a message in es/en/pt inside the chat. Once per session and desktop only.	Off



The proactive greeting bubble with its delay and its texts in Spanish, English and Portuguese.



The greeting bubble next to the button, without opening the chat.



On /checkout, the chat opens by itself.

Two rules so it isn’t annoying

- If the visitor **closes the chat**, it won’t reopen by itself for **30 seconds**, and whatever they already dismissed in that session isn’t repeated.

- On **mobile** (screens up to 480 px) the chat **never opens by itself**: bubbles do appear, and the visitor decides whether to tap them.

> **TIP**
> If the text in the visitor's language is missing, the bubble uses whichever exists, in this order: Spanish, English, Portuguese. Write all three if you have customers from several countries.

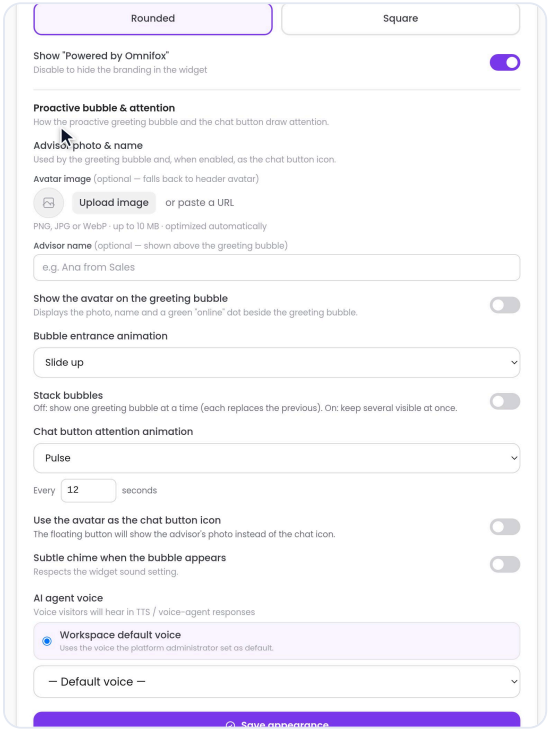
17

CHAPTER 17

Proactive bubble and advisor photo

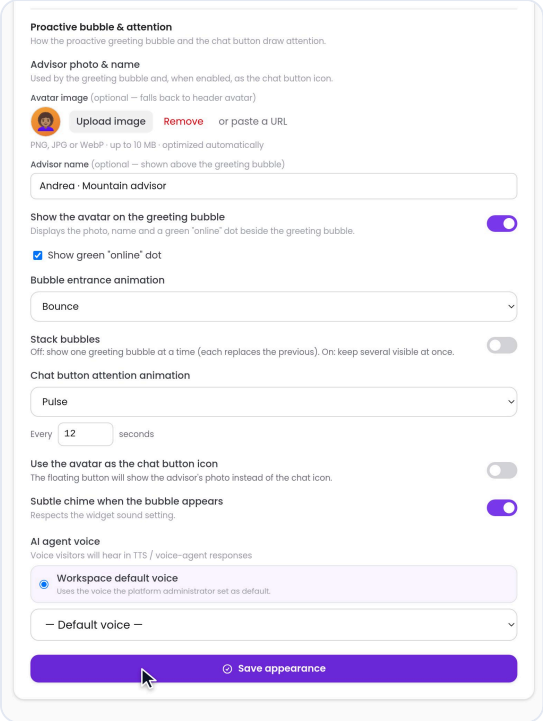
So the greeting feels like it's from a real person.

In **Appearance**, the **Proactive bubble and attention** block decides how greeting bubbles look, both those from chapter 16 and those from triggers.



Appearance → Proactive bubble and attention.

Option	What it does	Default / limits
Avatar image	The advisor's photo next to the bubble. Upload an image or paste a URL.	Empty = uses the header avatar · PNG, JPG or WebP up to 10 MB
Advisor name	Shown above the bubble (e.g. "Andrea · Mountain advisor").	Empty · max. 60
Show the avatar on the greeting bubble	Turns on the photo and the name.	Off
Show the green "online" dot	A green dot on the photo.	Ticked



The advisor's photo and name, the entrance animation and the bubble sound.

Option	What it does	Default / limits
Bubble entrance animation	How it appears.	Fade · Slide up (default) · Pop · Bounce
Stack bubbles	Off: one bubble at a time, each replaces the previous one. On: several visible, one above another.	Off
Subtle sound when the bubble appears	A soft sound. It also respects the sound setting in Availability.	Off
Use the avatar as the chat button icon	The advisor's photo replaces the button icon (chapter 9).	Off

Save with **Save appearance**.

TIP

Use the photo of a real person on your team, with their permission, and a name that matches whoever usually answers. A greeting from "Andrea" that is then answered by a bot without warning breeds distrust.

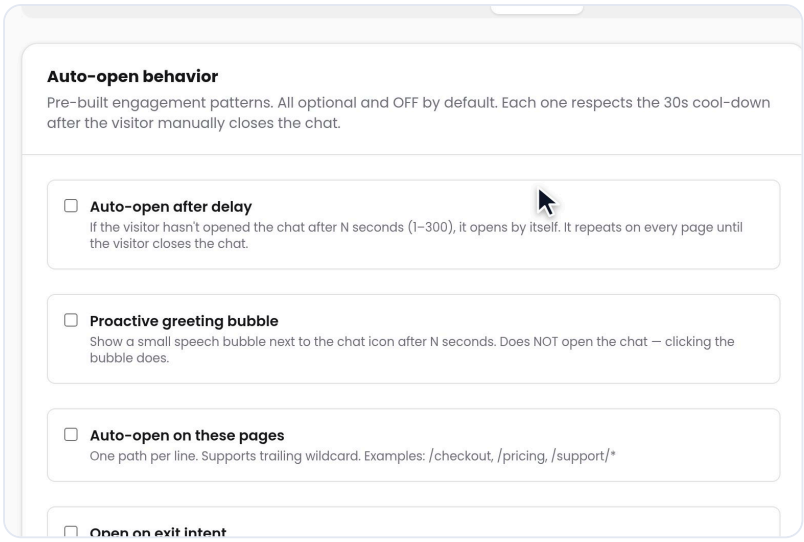
18

CHAPTER 18

Triggers: the first one step by step

When... Then...: your first rule, end to end.

Below auto-open are the **Behavioral triggers**: rules you build. Each one has a **When**, with one or more conditions, and a **Then**, with the action. They're saved with **Save triggers**.

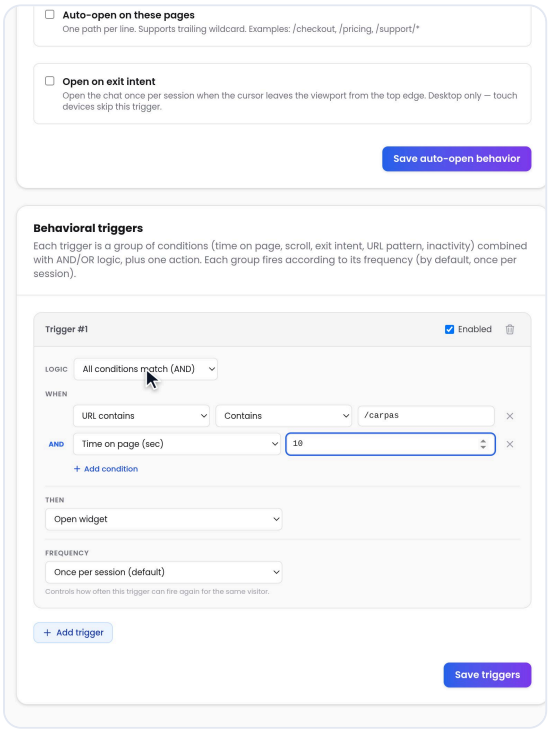


Behavioral triggers, still with none: Add trigger.

The goal of the example

When someone has been looking at the **tents** page for **10 seconds**, show them two greeting bubbles and, if they open the chat, have **Vela** (the AI agent) start the conversation to help them choose their tent.

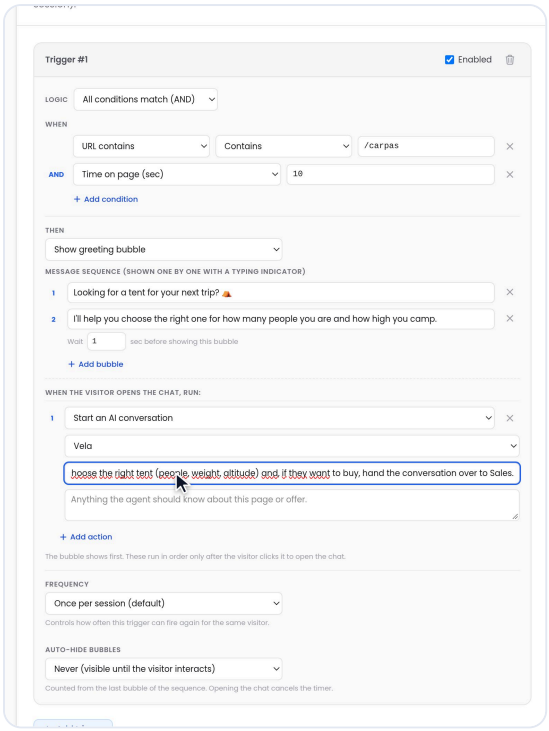
1. The When



When: the URL contains /carpas and 10 seconds on the page, with the All conditions logic.

1. Click **Add trigger**. **Trigger #1** appears, with its **Enabled** switch and the Delete button.
2. First condition: **URL contains** → `/carpas` (the path of the store's tents page).
3. **Add condition**: **Time on page (sec)** → `10`.
4. In **Logic**, leave **All conditions match (AND)**: both must be met.

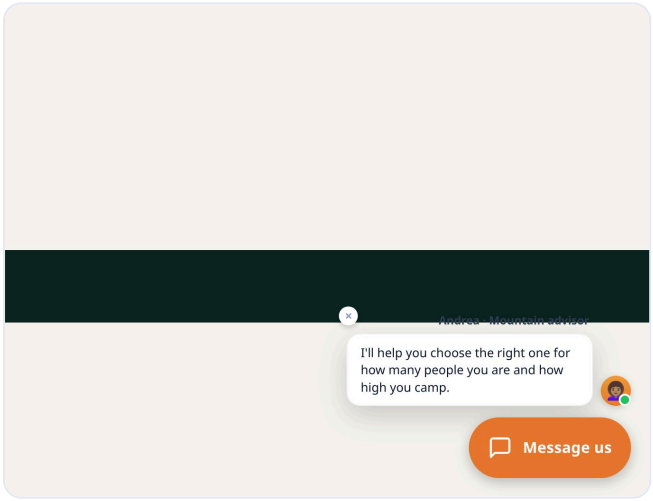
2. The Then



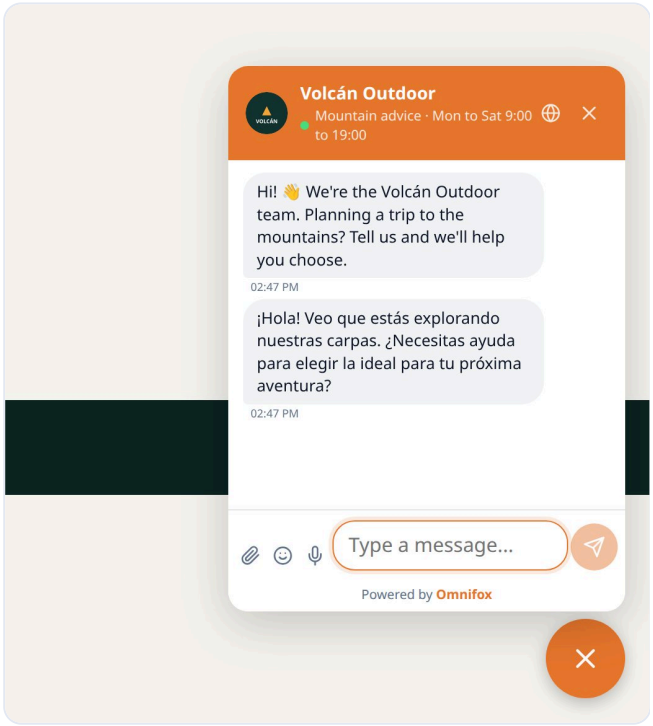
Then: a bubble split into two messages and, when the chat opens, start a conversation with Vela.

1. In **Then**, choose **Show greeting bubble**.
2. Tick **Split into a sequence of messages** and write two: “Looking for a tent for your next trip? ⛺” and “I’ll help you choose the right one for how many people you are and how high you camp.”. Each message can wait from 0 to 15 seconds, with a typing indicator.
3. In **When the visitor opens the chat, run:**, add **Start an AI conversation**, choose **Vela** and write the **Goal**: “Help choose the right tent (people, weight, altitude) and, if they want to buy, hand the conversation over to Sales.”.
4. Click **Save triggers**.

3. The test



On /carpas, after 10 seconds, the two bubbles appear.



When the chat opens, Vela is already typing and knows which page the visitor is on.

A visitor enters `/carpas`, waits, and after 10 seconds the two bubbles appear. When the chat opens, Vela is already typing, and she knows which page the visitor is on: she receives the address, the page title and where they came from.

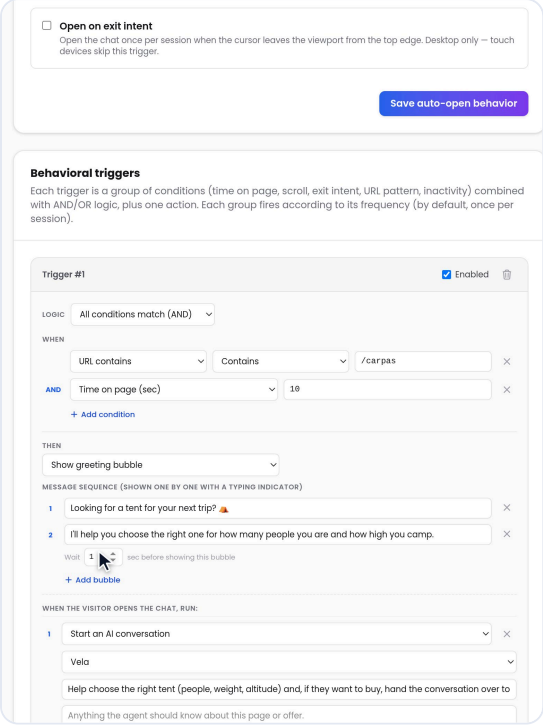
TIP

To test a "once per session" trigger, use a new incognito window each time: the widget remembers what it already showed.

CHAPTER 19

19 All the conditions and AND/OR logic

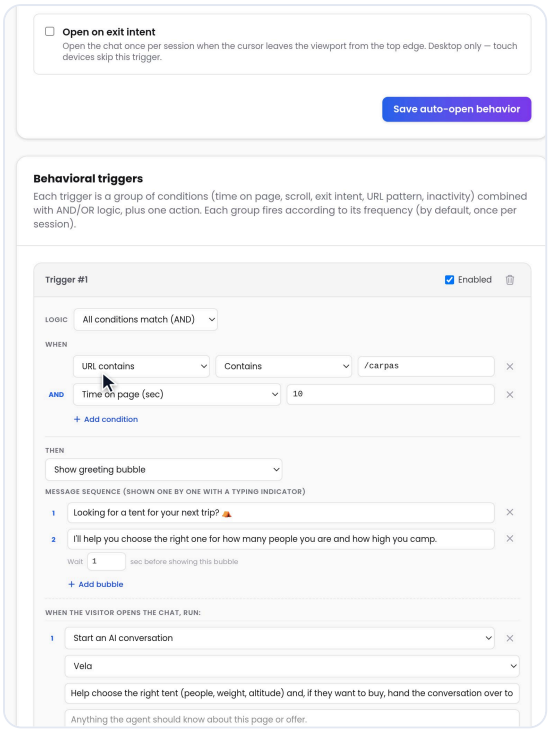
Eight conditions, four operators and two logics.



The When condition types.

Condition	Met when…	Value
Time on page (sec)	the visitor has been on the page for N seconds.	From 1
Scroll (%)	they scrolled down at least that percentage of the page.	1 to 100
Scroll (px)	they scrolled down at least that many pixels.	Pixels
Exit intent	the cursor leaves through the top of the window (desktop only).	—
URL contains	the page address matches the operator.	Max. 255 characters
Inactivity (sec)	they stop moving the mouse, scrolling or typing for N seconds.	From 5
Device	the visitor uses that type of device.	Desktop · Mobile · Tablet
The visitor opens the chat	the visitor opens the chat (by any means).	—

The address operators



The address operators and the All / Any condition logic.

Operator	Example	Matches
Contains	`carpas`	/carpas, /carpas/cotopaxi-2p, /ofertas-carpas
Is exactly	`/`	only the home page
Starts with	`/blog/`	every blog post
Does not contain	`checkout`	every page except checkout

Logic	What it means
All conditions match (AND)	All must be met.
Any condition matches (OR)	One is enough.

Up to 10 conditions

per trigger.

Up to 30 triggers

per widget.

TIP

The visitor opens the chat is for actions that should only happen once the person has shown interest: for example, “if they open the chat on /checkout, assign to Sales”.

NOTE

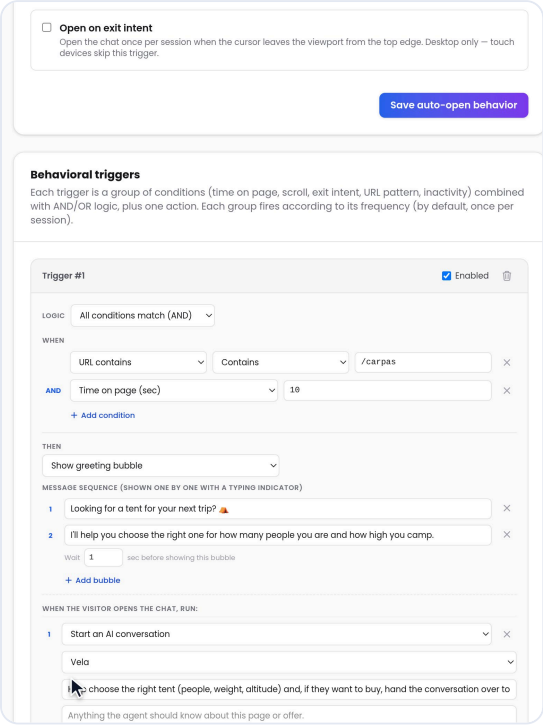
The minimum inactivity is 5 seconds; if you write less, saving shows an error saying inactivity must be at least 5 seconds.

20

CHAPTER 20

All the actions, frequency and hiding

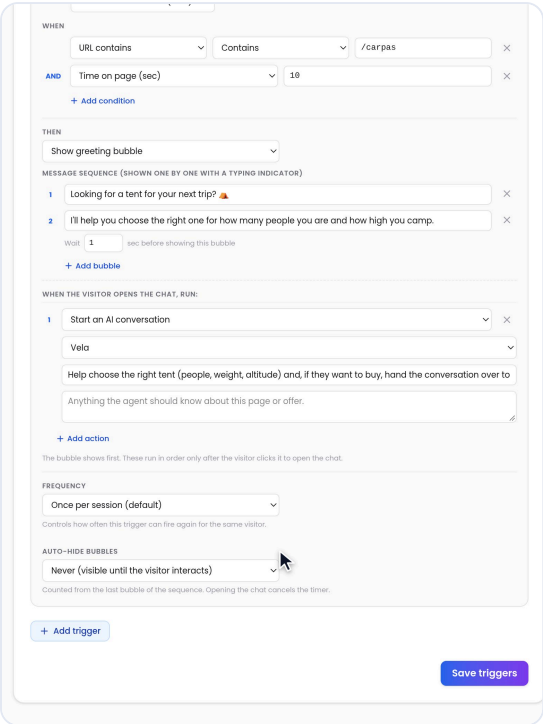
Nine actions, five frequencies and bubbles that retire by themselves.



The Then actions.

Then action	What it does
Open widget	Opens the chat.
Show greeting bubble	One bubble (max. 500 characters) or a sequence of up to 5 messages, with a 0 to 15 s wait between them.
Hide widget	Hides the button and the chat (closes it if open, and removes the bubble). Useful on pages where you don't want the chat.
Send a message	An agent message in the visitor's conversation, and optionally **Then assign to** someone.
Start an AI conversation	An AI agent writes first, with a **Goal** (max. 1000) and **Context / instructions** (max. 2000).
Assign the conversation	To a **Human agent**, a **Team** or an **AI agent**.
Run a workflow	Runs one of your active workflows on the visitor's conversation.
Show the departments selector	Shows the departments step (chapter 13).
Show the pre-chat form	Shows the form (chapter 12).

Frequency and hiding



Frequency and Auto-hide bubbles.

Option	What it does	Default / limits
Frequency	How often the trigger repeats.	Once per session (default) · On every page load · Every time it matches · Once a day · Only once per visitor
Auto-hide bubbles	Removes the bubbles if nobody touches them. Counted from the last bubble; opening the chat cancels it.	Never · After a while (0 to 600 s)

When the visitor opens the chat

Only in **Show greeting bubble**, the **When the visitor opens the chat, run:** block allows up to **10 actions** that run in order as soon as the visitor opens the chat, by any means (the bubble, the button or your own button with the API): send a message, start AI, assign, run a workflow, show departments or show the pre-chat.

> **NOTE**
> "Server" actions (send a message, start AI, assign, run a workflow) happen **only once per conversation**, even if the visitor reloads or the frequency allows repeating. Visual ones (open, bubble, hide) do follow the frequency.

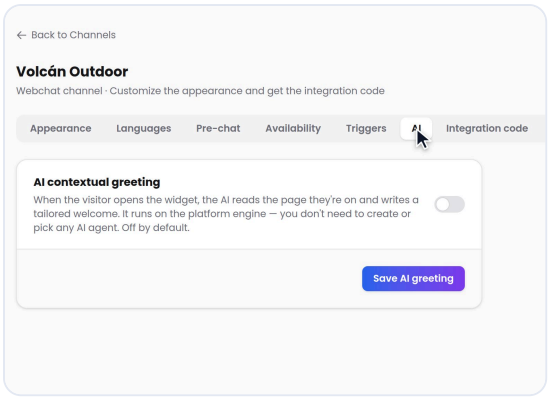
21

CHAPTER 21

AI contextual greeting

A welcome written by AI for the exact page the visitor is on.

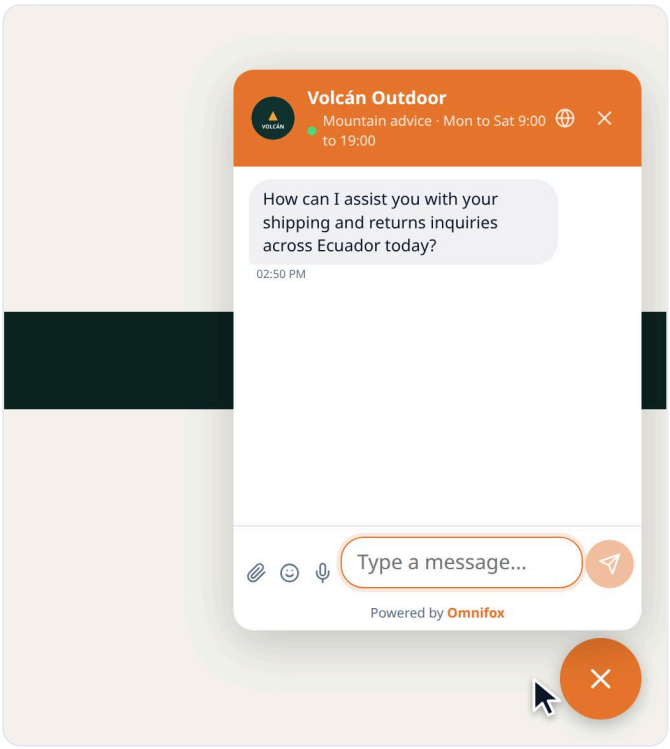
The **AI** tab holds the **AI contextual greeting**: when the visitor opens the chat, the AI reads the page they're on (address, title and an excerpt of the text) and writes a tailored one- or two-sentence welcome, with a typing indicator. **You don't need to create any agent**: it uses the platform's engine.



AI tab: AI contextual greeting and its cooldown.

Option	What it does	Default / limits
AI contextual greeting	Replaces the fixed welcome message when the visitor opens the chat.	Off
Cooldown	How often it greets the same person.	Once per session (recommended) · Once per hour · Once per day · Always

Save with **Save AI greeting**. The greeting comes out in the visitor's language; if the AI doesn't respond, a fallback in Spanish, English or Portuguese is used.



On the shipping page, the AI greeting talks precisely about shipping.

In the example, on the shipping page, the greeting talks precisely about shipping. If the visitor replies, that greeting is saved in the conversation, so your team sees what the AI told them.

TIP

Each greeting is a call to the AI. "Once per session" is recommended; with "Always", every chat opening generates a new greeting.

CAREFUL

If you have the AI greeting and also a trigger that starts an AI conversation when the chat opens (chapter 18), the visitor would get **two greetings**. Use one of the two on each page.

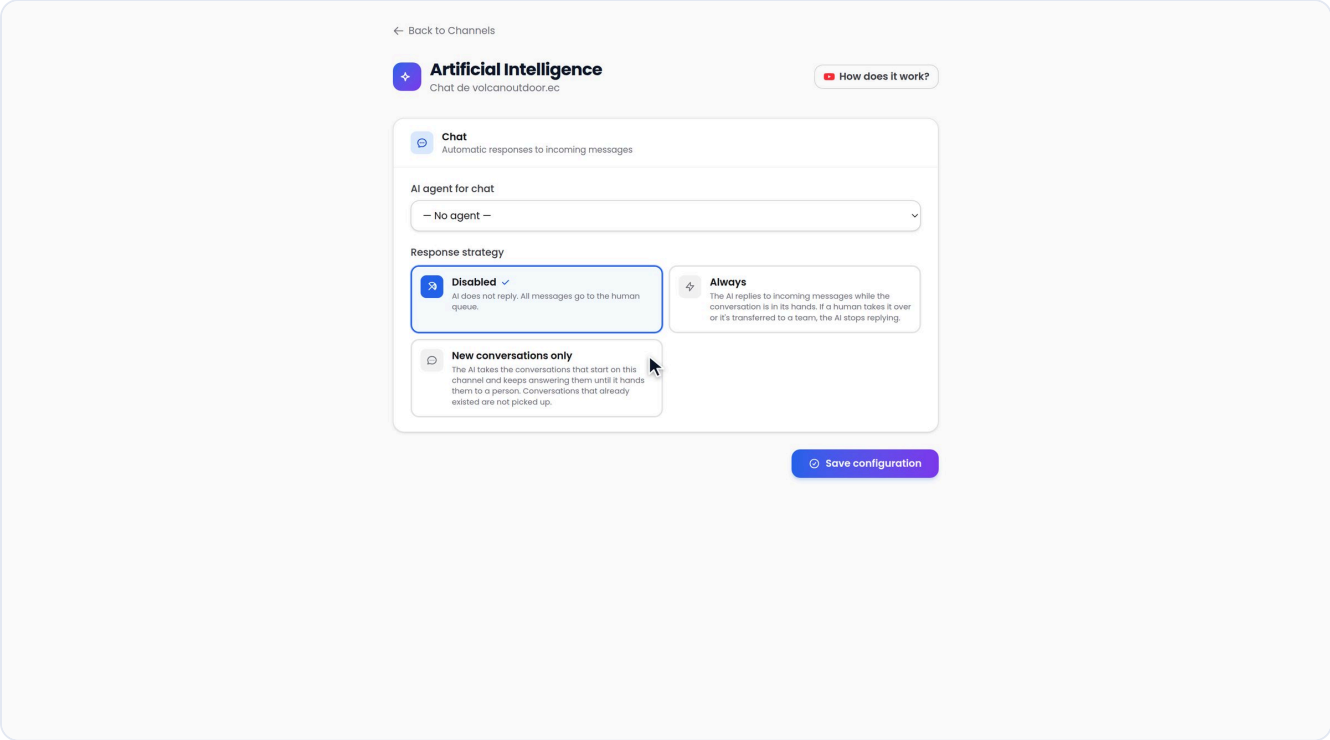
22

CHAPTER 22

The channel's AI agent

An AI agent that replies instantly and hands over to a person what matters.

For an AI agent to answer chat messages, open the channel's **Artificial Intelligence** screen (the AI button on the card, chapter 4).



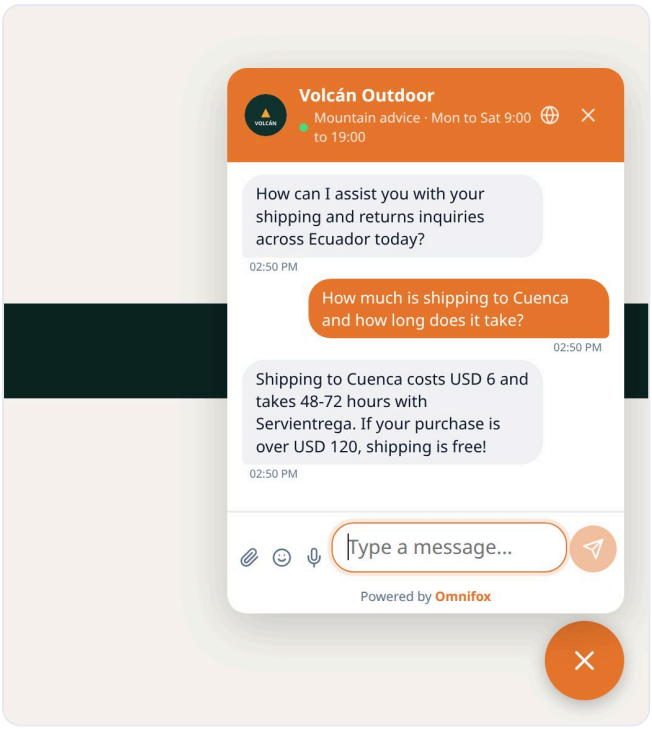
The channel's Artificial Intelligence: the agent Vela and the reply strategy.

Option	What it does	Values
AI agent for chat	The agent that answers on this channel. Only agents from the same workspace. If the agent is off, you'll see a notice and the **Open agent** link.	— No agent — (default) · your agents
Reply strategy	When the AI answers (table below).	Disabled (default) · Always · New conversations only

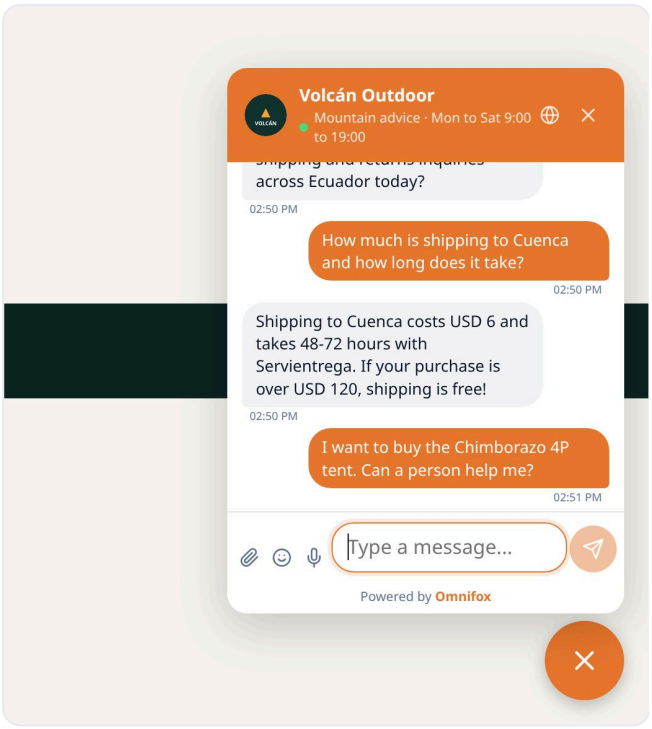
Strategy	What happens
Disabled	The AI doesn't answer. All messages go to the people's queue.
Always	The AI answers as long as the conversation is in its hands. As soon as a person takes it or it's transferred to a team, it stops answering.
New conversations only	Takes the conversations that start on this channel from that moment and keeps handling them until handing them to a person. Existing ones are left alone.

Save with **Save settings**. Below the chosen agent its schedule is shown: if the agent has its own schedule, outside it it doesn't answer even if the strategy says Always.

How Vela works



Vela replies with the store's real details.



The visitor asks to talk to a person: Vela hands the conversation over.

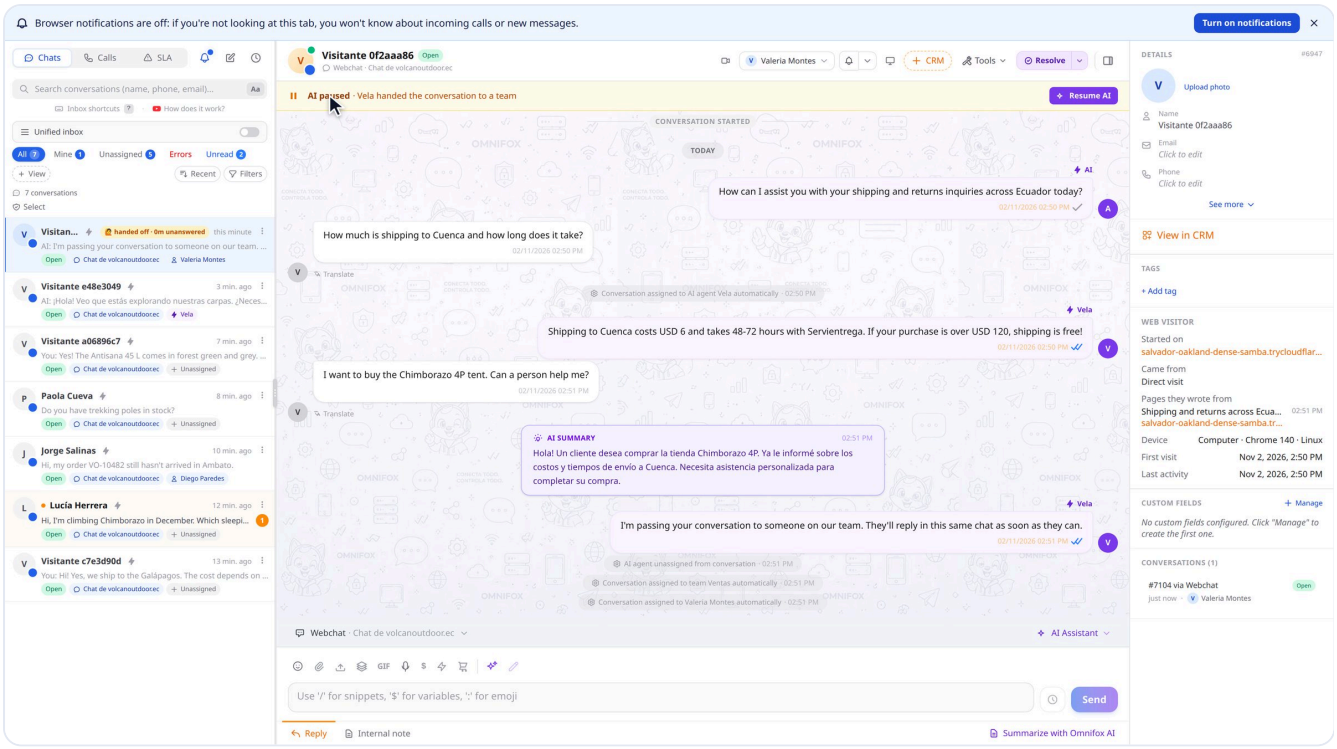
How much is shipping to Cuenca and how long does it take?

Shipping to the rest of the country costs USD 6.00 and arrives in 48 to 72 hours. If your order is USD 120 or more, it's free. 🚚

I want to buy the Chimborazo 4P tent. Can a person help me?

I'm passing your conversation to someone on our team. They'll reply in this same chat as soon as possible.

Vela transferred the conversation to the Ventas team, with an AI summary; the AI is paused



In the inbox: the AI paused after Vela transferred the conversation to a team, the AI summary and the conversation assigned to Ventas.

> **NOTE**
>
> How to create the agent, give it your business knowledge and decide when it transfers is in the complete guide to **AI Agents**.

23

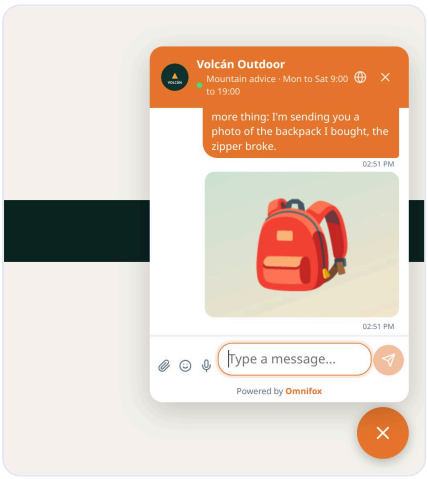
CHAPTER 23

What the visitor can do

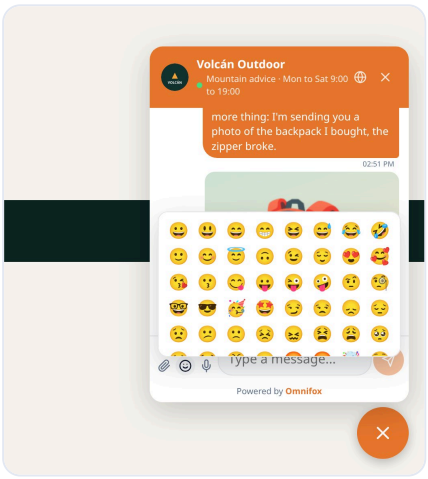
Attachments, emojis, voice notes, formatting, typing and read.

On the visitor's side, the chat has everything they expect from modern messaging:

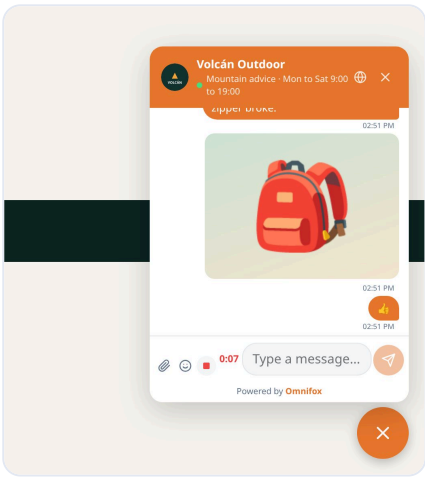
Feature	Detail
Attach (paperclip)	Photos, PDFs, audio, video and Word, Excel, PowerPoint, OpenDocument, TXT or CSV documents, several at once, **up to 20 MB** each. Images show as thumbnails and enlarge with a click. SVG and HTML aren't accepted.
Emojis	The widget's own emoji panel.
Voice note (microphone)	Up to **3 minutes**. It reaches the inbox as audio. The browser asks permission to use the microphone.
Links and formatting	Links become clickable. Your messages can include `**bold**`, `*italics*`, `` `code` `` and lists with "- ".
Reply buttons	Questions with options from your workflows and the AI show as buttons.
Typing...	The visitor sees when your agent is typing, and your agent sees when the visitor is typing.
Delivered and read	Each message shows when it was delivered and when it was read. Each message's time is relative: "1 minute ago", "yesterday".



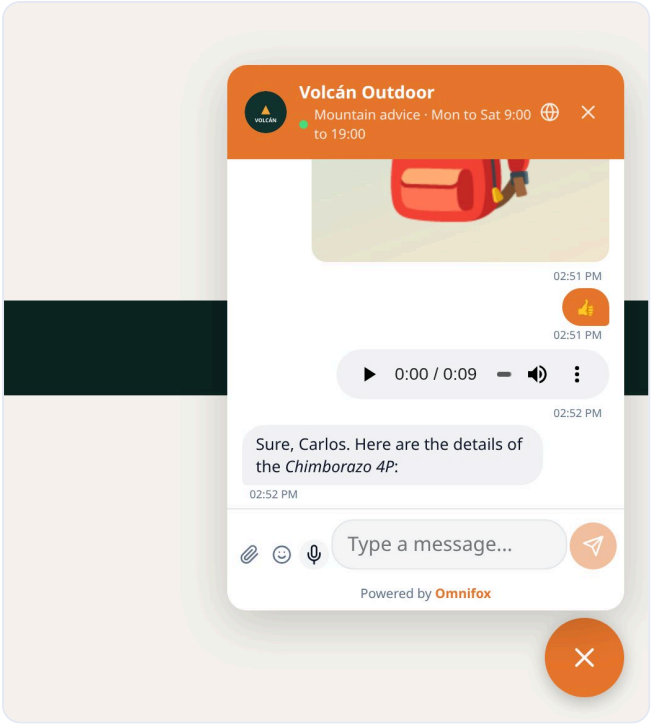
An attached photo, with its thumbnail.



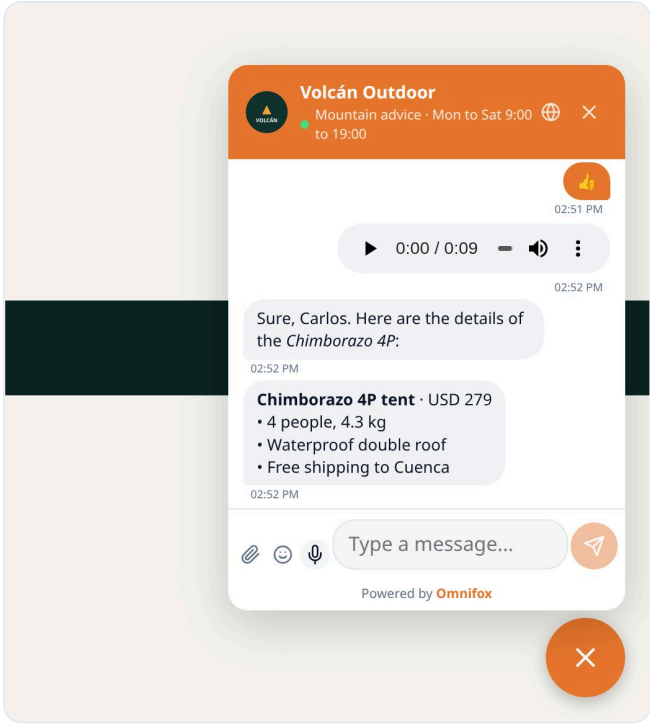
The widget's emoji panel.



Recording a voice note.



The notice that the agent is typing.



A message with bold and a list.

If something isn't sent

- A file over 20 MB isn't uploaded: the visitor sees that the file exceeds the 20 MB limit and wasn't sent.
- A type that isn't allowed: the visitor sees that this type of file can't be sent.
- If a text or file fails (for example, offline), the bubble fades with a red line and the **Retry** button.
- If the browser doesn't allow the microphone: "We couldn't use the microphone. Allow access in your browser to send voice notes.".

CAREFUL

In the Webchat, `*text*` is *italics* and `**text**` is **bold**. In WhatsApp, on the other hand, one asterisk is already bold. If your team writes "WhatsApp style", on the web chat they'll see italics.

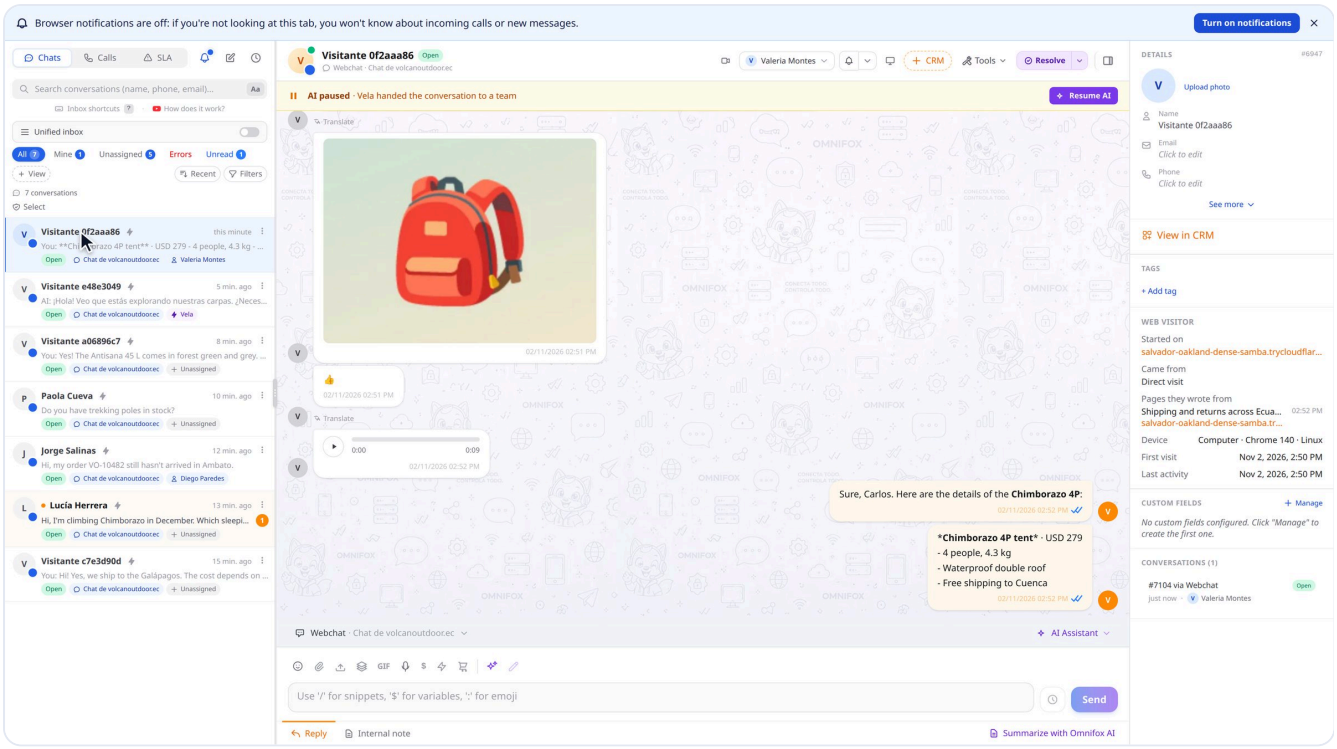
24

CHAPTER 24

Handle it from the inbox

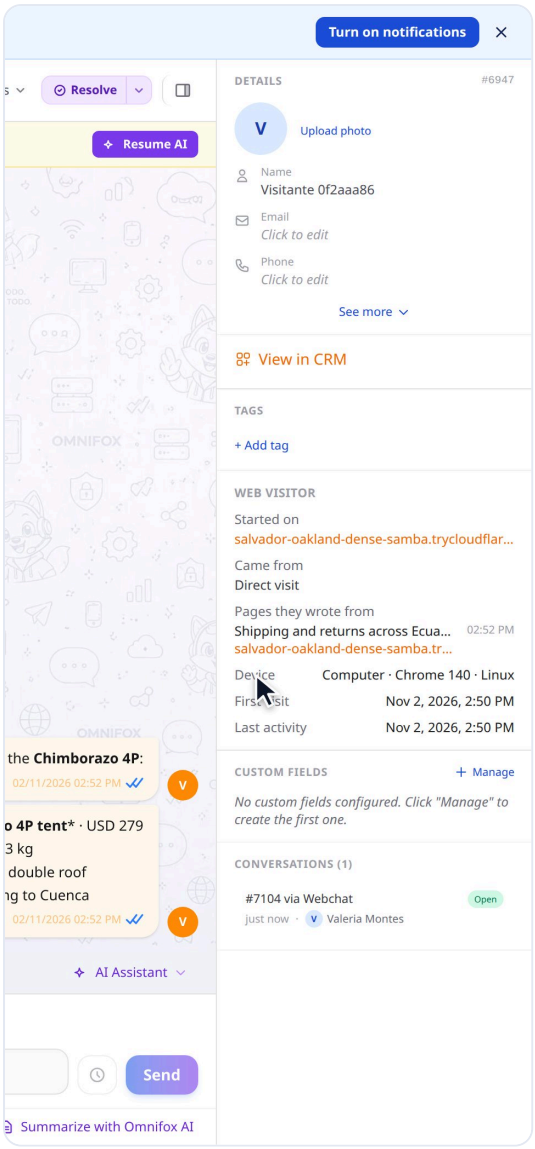
Webchat conversations, alongside everything else.

In the inbox, Webchat conversations carry the web chat icon and the channel name. The **green dot** means the visitor is **online right now**, with your site open.



The Webchat conversation in the inbox, with the green “online” dot.

The contact panel



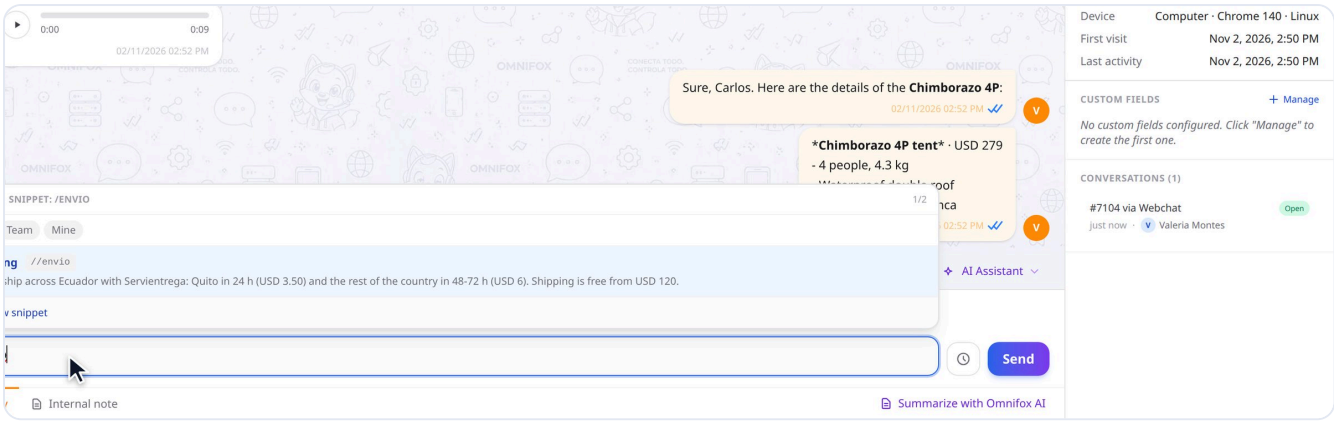
The Web visitor panel: where they started, where they came from, which pages they wrote from and their device.

Section	What it shows
Web visitor → Started on	The title and address of the page they first wrote from.
Came from	The source page (Google, a social network, another site) or “Direct”.
Pages they wrote from	The pages they sent messages from, most recent first (up to 10).
Device	Type, browser and system.
First visit · Last activity	When they started and when they were last seen.
Pre-chat	The answers to the form’s free questions and the chosen department. Name, email, phone and custom fields are in the contact details.

> **NOTE**
>
> The Web visitor panel only shows the pages the visitor **wrote** from, not everything they browsed. Watching visitors browse live is part of Co-browse (chapter 30).

Reply

You reply as on any channel: text, attachments, **quick replies** (type `/` and its name, like `/envio`), **internal notes** the visitor doesn’t see, or **assigning** to a colleague or a team. Everything reaches the visitor instantly.



The /envio quick reply in the composer.

When the visitor leaves

If the visitor closes the tab, the dot turns off and the header shows when they were last seen. Your reply still reaches them: they'll see it as soon as they come back to your site **in that same browser** (with counter and sound, chapter 15). If the visitor left their email, you can also write to them through your Email channel.

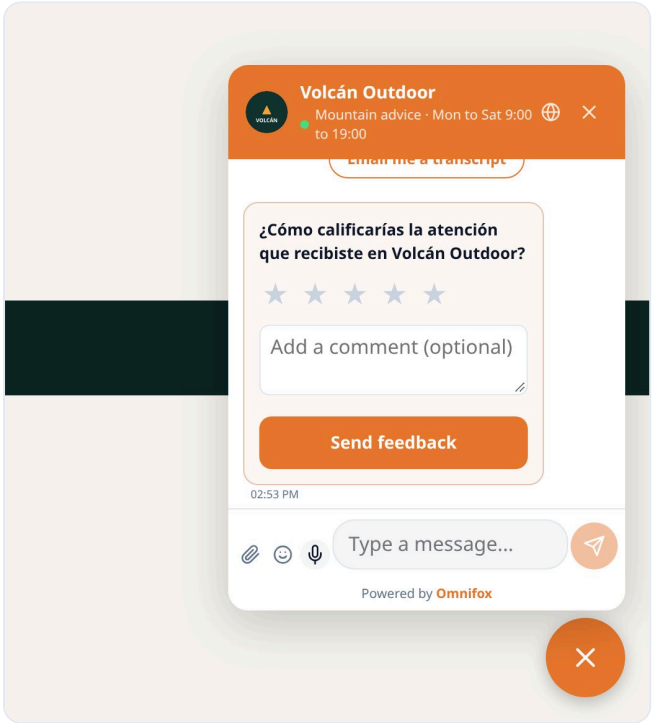
25

CHAPTER 25

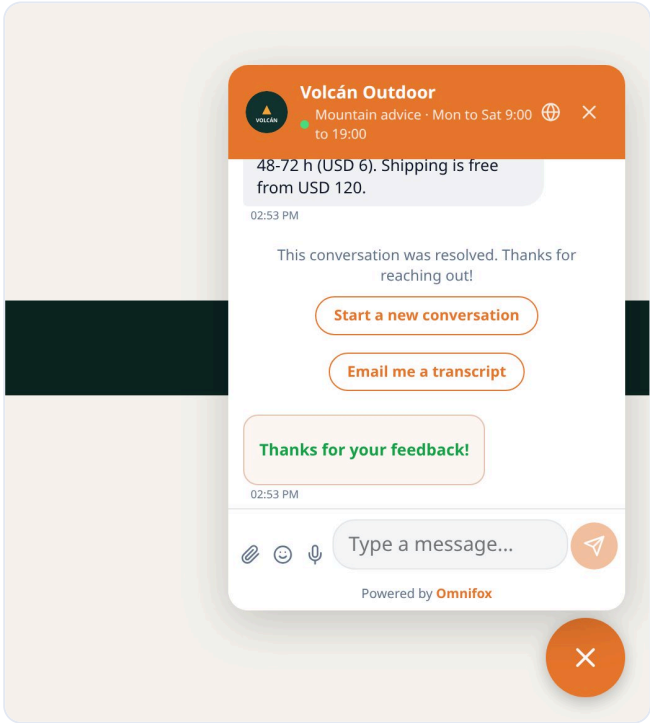
Resolve, survey and transcript

Close well: resolved notice, rating and a copy by email.

When the case is done, mark the conversation as **resolved** (or close it). The visitor sees a divider: "This conversation was resolved. Thanks for reaching out!" (or a closed-by-the-agent notice).



Conversation resolved: the visitor rates the service inside the chat.



After rating: the thank-you, the email transcript and Start a new conversation.

Satisfaction survey

If your workspace's **satisfaction survey** is on (Settings → Satisfaction survey), on resolving it arrives as a message and the visitor **rates right there**, inside the chat, in your survey's format (stars, emojis, thumbs or numbers) with a comment. The rating reaches the conversation and your reports.

Email transcript

In the **Pre-chat** tab, the **Email transcript** block (**on** by default) offers the visitor, when the conversation closes, the option to receive the conversation by email: they type their email, tap **Send** and get a copy of the chat. If they give a new email, it's saved on their contact.

> **CAREFUL**
> The copy goes to the email the visitor types: it has to be a real email of theirs. If you turn it off, the option disappears from the chat and sending is blocked.

A new conversation

With **Start a new conversation**, the visitor begins another one whenever they want. And if they simply write again shortly after, the same conversation reopens (chapter 26).

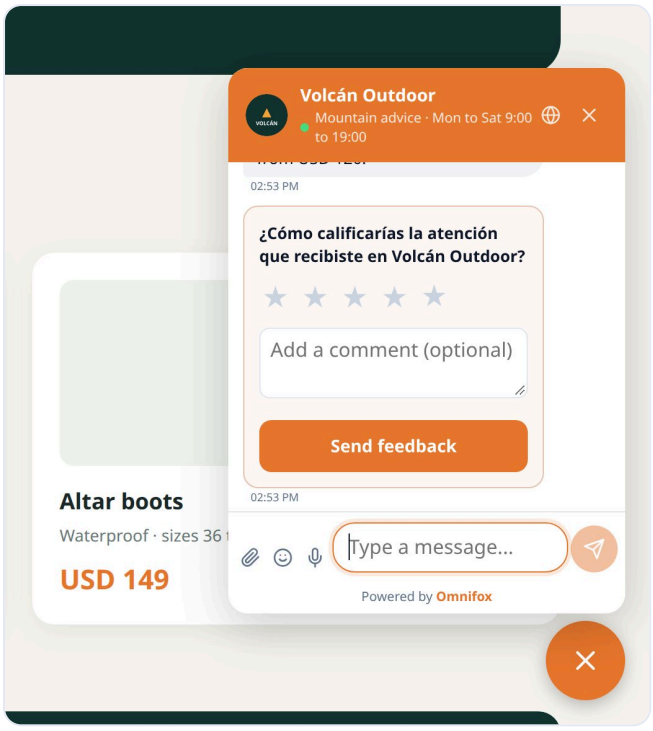
26

CHAPTER 26

The returning visitor

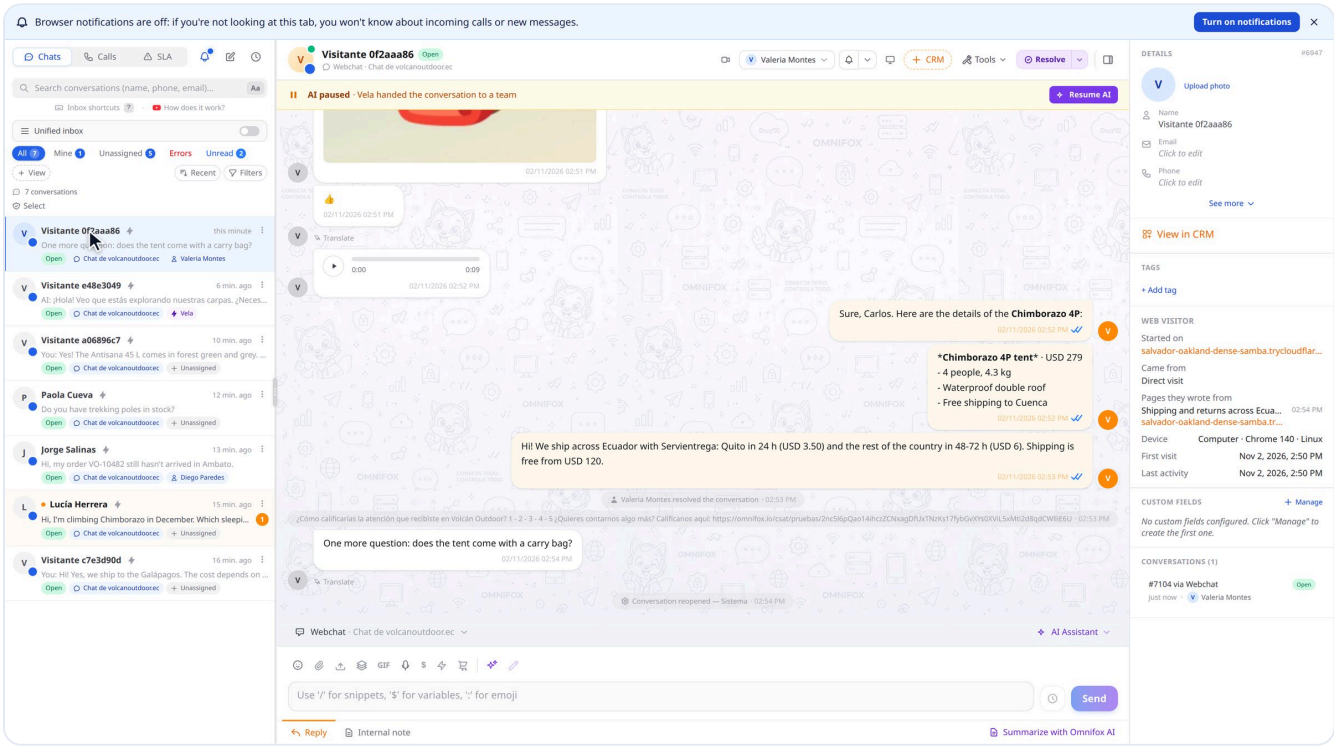
The chat remembers the visitor in that browser.

The Webchat remembers the visitor **in that browser**. If they reload, change page or come back later, they see their conversation just as they left it, with all the history.



Back in the store, the conversation is just as it was left.

If the visitor…	This happens
reloads or changes page	The chat stays the same, with its history.
comes back another day to the same browser	They see their conversation; the welcome isn't repeated.
writes shortly after it was resolved	The **same conversation reopens**, within your workspace's grace period (240 minutes by default).
writes much later	A new conversation starts, with the **same contact** and all their history.
uses another browser or device	They're a new visitor, unless your site identifies them (chapter 28).



The new message reopens the same conversation, with the same contact.

Shared computers and switching accounts

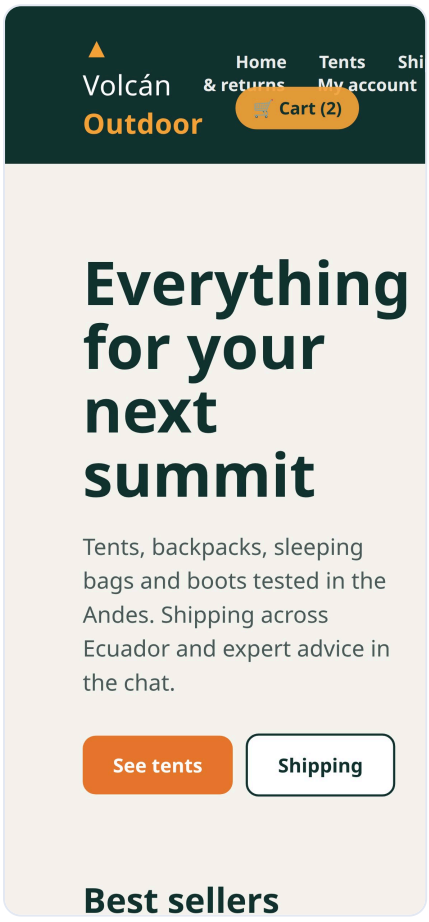
When a customer signs out of your site, or on a shared computer, it's best to call `Omnifox.logout()`: the widget forgets the previous visitor (conversation, history and identity) and the next one starts from scratch.

```
// When signing out of your site (before redirecting)
document.getElementById('sign-out').addEventListener('click', function () {
  if (window.Omnifox && Omnifox.logout) Omnifox.logout();
});
```

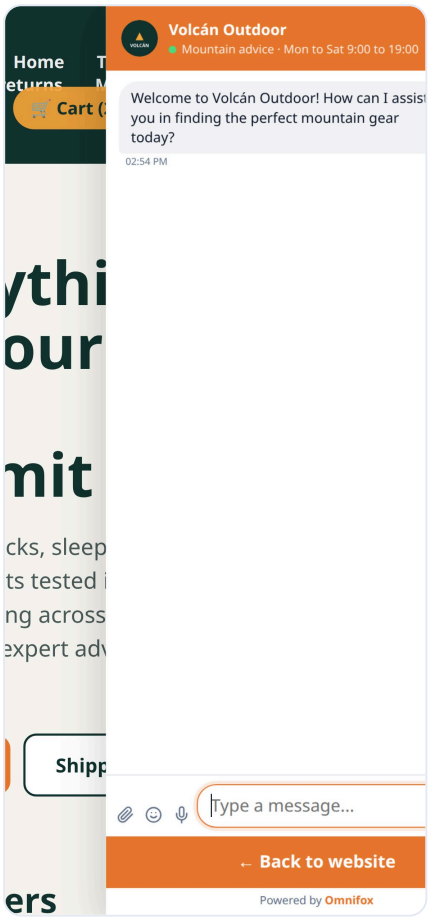
CHAPTER 27

27 On mobile

Full screen, keyboard and a button to go back to the site.



The store on mobile, with the chat button.



The chat in full screen.



Back to website closes the chat.

- On mobile (screens up to 480 px wide), the chat opens **full screen** and adjusts to the keyboard, on iPhone too.
- At the bottom it has a big ← **Back to website** button, so nobody gets stuck if they opened it by accident. The phone's back button works too.
- The text next to the button can show only on mobile or only on desktop (chapter 9).
- On mobile the chat **never opens by itself**: bubbles do appear, and the visitor decides whether to tap them (chapter 16). Exit intent doesn't exist on mobile.

TIP

The editor preview shows the desktop chat. After big changes, open your site from your phone too to see how it looks.

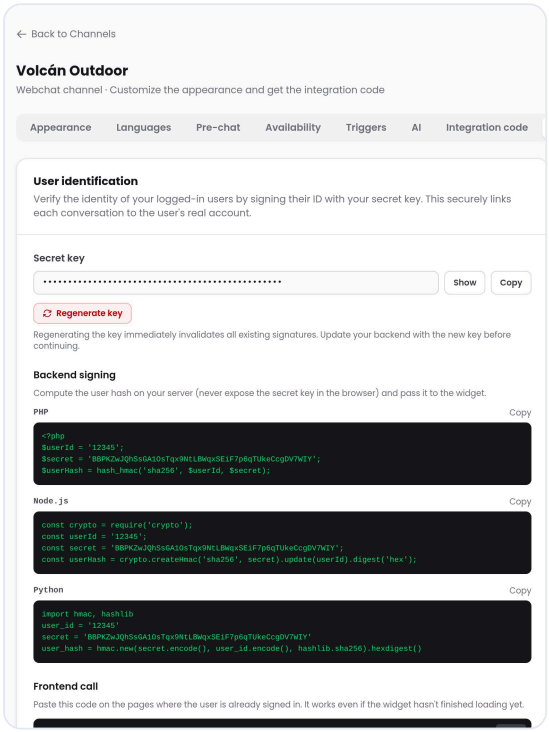
28

CHAPTER 28

Secure user identification

Recognize your signed-in customers, with a signature nobody can forge.

If your site has customers who sign in, the **User identification** tab lets you recognize them securely: your **server** signs the user's identifier with a secret key, and your **page** passes that identifier, the signature and their details to the widget. That way the conversation arrives under your customer's name, linked to their record, even if they change computers.



User identification: the secret key with Show, Copy and Regenerate key.

Option	What it does	Default / limits
Secret key	Generated automatically the first time an administrator opens the tab. **Show**/**Hide** and **Copy**.	48 characters · only visible to those who can manage channels
Regenerate key	Creates a new key, with confirmation. Previous signatures **stop working instantly**.	—
Backend signature	Examples in PHP, Node.js and Python.	—
Frontend call	The snippet with `Omnifox.identify`, with a command queue.	—

How it works

1. Your server → Computes `user_hash = HMAC-SHA256(user_id, secret key)`, in hexadecimal (64 characters).

2. Your page → Calls `Omnifox.identify` with `user_id`, `user_hash` and the customer's details.

3. Omnifox → Checks the signature and links the conversation to the contact with that identifier (or adopts the contact with the same email that didn't have an identifier yet).

1. The signature, on your server

PHP

```php
<?php
// The secret key lives on your server (environment variable), never in the browser
$secret  = getenv('OMNIFOX_IDENTIFY_SECRET');   // e.g. "your_48_character_secret_key"
$userId  = (string) $user->id;                  // your internal identifier, as text
$userHash = hash_hmac('sha256', $userId, $secret);  // 64 hexadecimal characters
```

NODE.JS

```node
// Node.js
const crypto = require('crypto');
const secret = process.env.OMNIFOX_IDENTIFY_SECRET;          // never in the browser
const userId = String(user.id);
const userHash = crypto.createHmac('sha256', secret).update(userId).digest('hex');
```

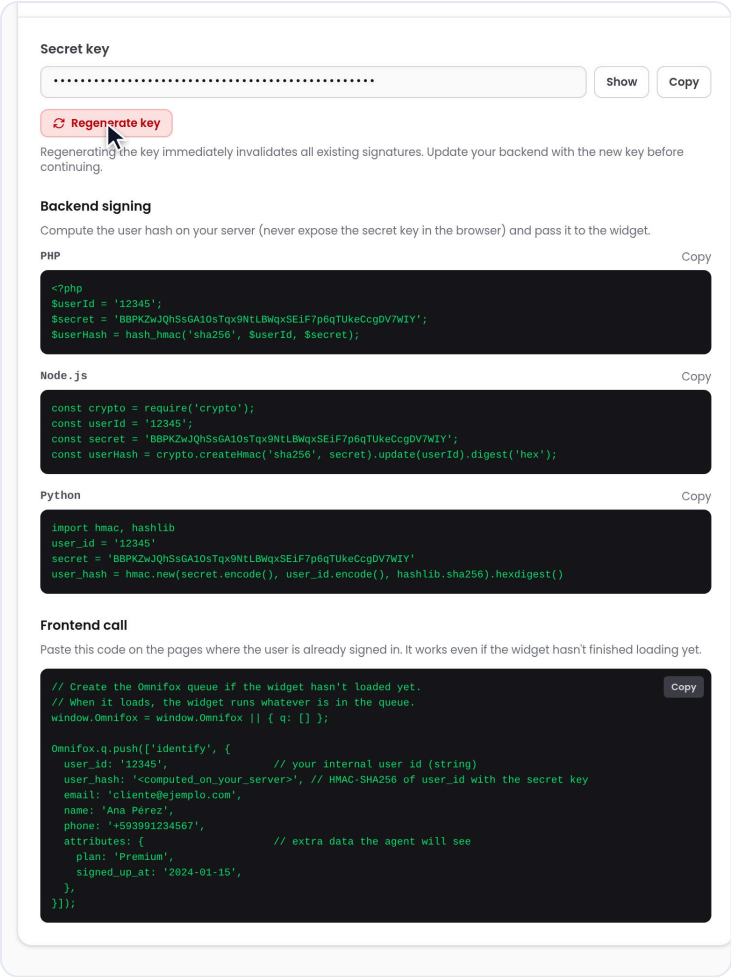
PYTHON

```python
# Python
import hmac, hashlib, os
secret = os.environ["OMNIFOX_IDENTIFY_SECRET"]               # never in the browser
user_id = str(user.id)
user_hash = hmac.new(secret.encode(), user_id.encode(), hashlib.sha256).hexdigest()
```

2. The call, on your page

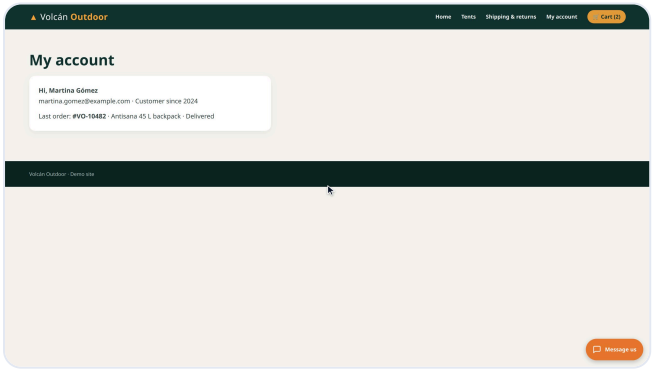
```html
<script>
  // Works even if the widget hasn't loaded yet (command queue)
  window.Omnifox = window.Omnifox || { q: [] };
  Omnifox.q.push(['identify', {
    user_id:   'VO-2291',                       // the same text you signed
    user_hash: '<computed_on_your_server>',     // the HMAC-SHA256 in hexadecimal
    name:  'Martina Gómez',
    email: 'martina.gomez@example.com',
    phone: '+593 99 000 0000',
    attributes: { customer_since: '2024', last_order: 'VO-10482' }
  }]);
</script>
```

Field	Required	What it is
`user_id`	Yes	Your internal customer identifier, as text. Exactly the same one you signed.
`user_hash`	Yes	The HMAC-SHA256 signature computed on your server.
`name`, `email`, `phone`	No	Saved on the contact record, verified.
`attributes`	No	Extra data (customer since, last order...).

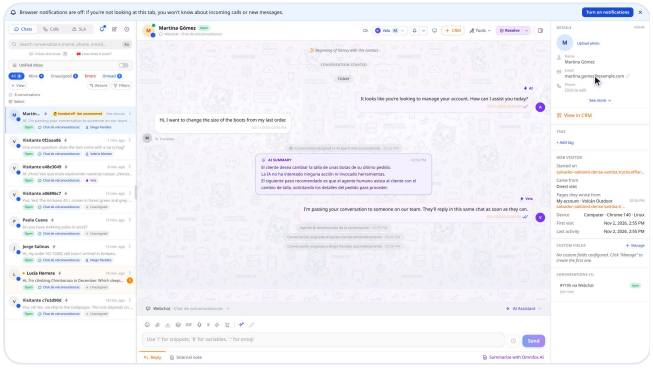


The backend signature (PHP, Node.js, Python) and the frontend call.

In the store



The store's "My account" page, with the signed-in customer.



The conversation arrives under Martina Gómez, with her verified email.

Martina signs in and opens **My account**; the store's server computes her signature and the page identifies her. She writes "Hi, I want to change the size of the boots from my last order.", and in the inbox the conversation arrives under **Martina Gómez**, with her email, linked to her customer record.

> **CAREFUL**
>
> The secret key **never** goes in the browser, in the page code or in a public repository: keep it in an environment variable on your server. If you suspect it leaked, use **Regenerate key** and update your server immediately (meanwhile, identifications fail and customers chat as anonymous visitors).

TIP

When the customer signs out, call `Omnifox.logout()` (chapter 26), so the next person using that computer doesn't inherit their conversation.

CHAPTER 29

29 JavaScript API and attributes

Open the chat from your buttons, tune each page and measure in GA4.

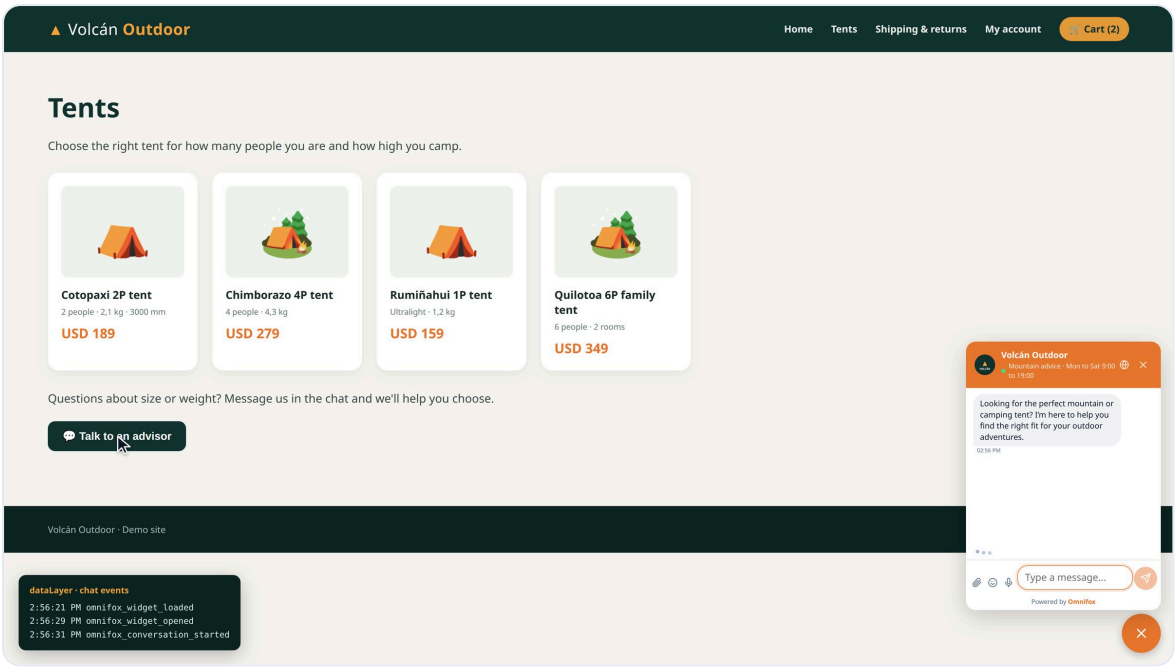
The window.Omnifox methods

```
Omnifox.open();      // opens the chat
Omnifox.close();     // closes it
Omnifox.toggle();    // opens it if closed and closes it if open
Omnifox.isOpen();    // true or false
Omnifox.identify({ user_id: '…', user_hash: '…', name: '…', email: '…' });
Omnifox.logout();    // forgets the visitor: conversation, history and identity
```

Method	What it does	Returns
`Omnifox.open()`	Opens the chat (and runs your triggers' "when the chat opens" actions).	—
`Omnifox.close()`	Closes the chat.	—
`Omnifox.toggle()`	Opens it if closed and closes it if open.	—
`Omnifox.isOpen()`	Tells whether the chat is open.	`true` / `false`
`Omnifox.identify(data)`	Identifies the customer with their signature (chapter 28). Without `user_id` and `user_hash` it does nothing.	—
`Omnifox.logout()`	Clears the visitor's session: conversation, history and identity.	—

For example, the store's **Talk to an advisor** button opens the chat like this:

```
<button type="button" onclick="Omnifox.open()">Talk to an advisor</button>
```



The store's "Talk to an advisor" button opens the chat with Omnifox.open().

The command queue

The widget code loads **asynchronously**: if your page calls `Omnifox.open()` as soon as it loads, the widget may not exist yet ("Omnifox is not defined"). There are three safe ways:

- the **command queue**: declare `window.Omnifox = window.Omnifox || { q: [] }` and queue each call as `[method, ...arguments]`; the widget runs them on load, and those you queue afterwards run instantly;
- wait for the `omnifox:api-ready` event, which the widget emits when the API is ready;
- or wait for the browser's `load` event.

```
<script>
  window.Omnifox = window.Omnifox || { q: [] };
  // Each entry is [method, ...arguments]; it runs when the widget finishes loading
  Omnifox.q.push(['open']);
</script>
<script>
  // Or: wait until the API is ready
  window.addEventListener('omnifox:api-ready', function () {
    document.getElementById('talk-to-advisor').onclick = function () { Omnifox.open(); };
  });
</script>
```

The code attributes

The line of code accepts `data-*` attributes to change the widget **on that page only**. They win over what you saved in the editor.

Attribute	Example	What it does
`data-widget-key`	`"YOUR_WIDGET_KEY"`	**Required.** Your channel key.
`data-color`	`"#0ea5e9"`	Overrides the primary color.

Attribute	Example	What it does
`data-position`	`"bottom-left"`	Overrides the position (`bottom-right` / `bottom-left`).
`data-name`	`"Support"`	Overrides the widget name.
`data-greeting`	`"Hi!"`	Overrides the welcome message.
`data-locale`	`"en"`	Forces the language (`es` , `en` , `pt`). Wins over everything else.
`data-auto-detect`	`"false"`	Turns off browser language detection.
`data-proactive-greeting`	`"Hi Ana, can I help with your order?"`	A custom greeting bubble for this page (the text is shown as is).
`data-proactive-key`	`"user_42"`	With this key, the bubble above shows only once for that value.
`data-proactive-delay`	`"5"`	Seconds before that bubble (0 to 120; 5 by default).
`data-visitor`	`'{"name":"Ana","email":"ana@example.com"}'`	Visitor details **for display only**, unverified (`name` , `email` , `phone` , `external_id`).

```
<script src="https://omnifox.io/widget/loader.js"
        data-widget-key="YOUR_WIDGET_KEY"
        data-locale="en"
        data-color="#0ea5e9"
        data-proactive-greeting="Hi! Need help choosing a tent?"
        data-proactive-delay="5"
        async></script>
```

window.omnifoxSettings

Instead of `data-visitor` , you can declare the visitor's details before the widget code:

```
<script>
  // Before the widget code: visitor details to display (not verified)
  window.omnifoxSettings = {
    visitor: { name: 'Ana Pérez', email: 'ana@example.com', phone: '+593…', external_id: 'customer-42' }
  };
</script>
```

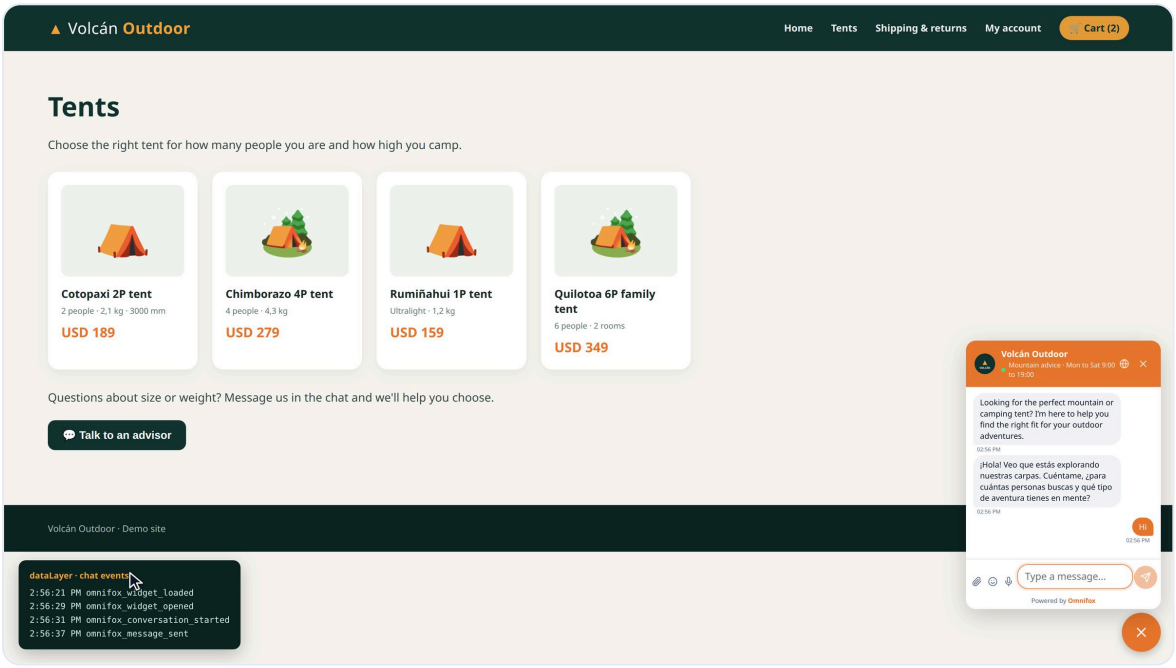
> **CAREFUL**
>
> `data-visitor` and `omnifoxSettings` **don't verify** anyone: anyone could type another name. To really identify customers, use `Omnifox.identify` with a signature (chapter 28).

Events for Google Analytics 4 and Tag Manager

The widget sends four events to your page's `dataLayer` (and to `gtag` , if you have it), each with the `omnifox_widget_key` parameter:

Event	When it's sent
`omnifox_widget_loaded`	The widget loaded on the page.
`omnifox_widget_opened`	The visitor opened the chat.
`omnifox_conversation_started`	A conversation started.
`omnifox_message_sent`	The visitor sent a message.

```
// What the widget pushes to your page's dataLayer (and to gtag, if present)
dataLayer.push({ event: 'omnifox_widget_loaded',        omnifox_widget_key: '…' });
dataLayer.push({ event: 'omnifox_widget_opened',        omnifox_widget_key: '…' });
dataLayer.push({ event: 'omnifox_conversation_started', omnifox_widget_key: '…' });
dataLayer.push({ event: 'omnifox_message_sent',         omnifox_widget_key: '…' });
```



The omnifox_* events the widget sends to the page's dataLayer.

- **GA4 with gtag.js**: the events appear by themselves in Reports → Events; mark `omnifox_conversation_started` as a **key event** to measure how many conversations start in the chat.
- **Google Tag Manager**: create a **Custom event** trigger with the event name (for example `omnifox_conversation_started`) and use it in your GA4 or ads event tag.
- With the funnel *loaded* → *opened* → *started* → *sent* you know how many visits turn into conversations and, crossing it with your purchases, how many sales start in the chat.

> **NOTE**
> Omnifox doesn't have a Webchat analytics screen for your account: these events are how you measure it. Conversations, response times and ratings are in the inbox reports.

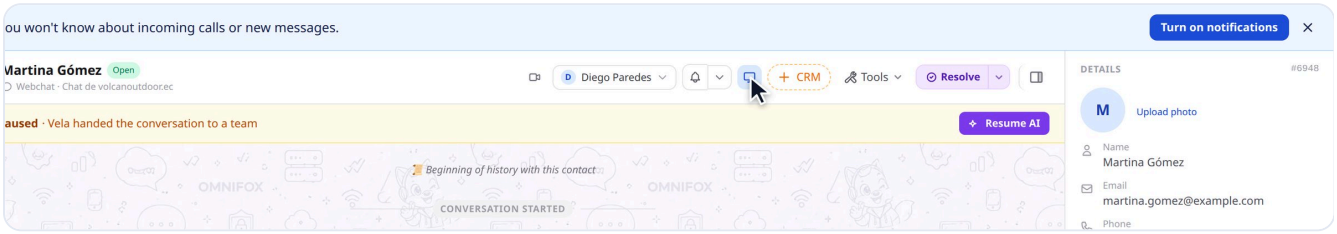
30

CHAPTER 30

Co-browse

See the visitor's screen and guide them, with their permission.

Webchat conversations have, in the header, the **Co-browse** button: with the visitor's permission, you see their screen inside your site and guide them (for example, to complete a purchase or a form). It also includes a radar of the visitors browsing your site right now.



The Co-browse button in the header of a Webchat conversation.

> **NOTE**
>
> Co-browse is an **add-on** with its own complete guide (video and manual): "Co-browse · Complete guide", in the Help Center and on the Omnifox YouTube channel. Without the add-on, the button doesn't appear.

31

CHAPTER 31

Good practices

What works best on sites that use the Webchat.

- **A short, concrete greeting.** A question that invites a reply ("Looking for a tent for your next trip?") is worth more than a welcome paragraph.
- **Trigger only where it helps**: the cart, pricing, an expensive product page. A chat that opens by itself on every page scares people away.
- **A form with just the details you need**: name and one way to contact them. The rest, while chatting.
- **Honest hours and away message**, with email capture if nobody is around at night.
- **An AI agent** that answers instantly and **hands over** to a person what matters (buying, complaining).
- **Department names for the visitor** ("Buying advice"), not internal ones ("Ventas").
- **Identify your signed-in customers** and call `Omnifox.logout()` on sign-out.
- **Measure** with the GA4 events and review your ratings.
- **Check Detected installations from time to time**, to know where your chat is.
- **Always test in incognito** and from your phone after big changes.

CHAPTER 32

32 Troubleshooting

What to check when something doesn't go as expected, FAQs and glossary.

Symptom	What to check
The chat doesn't appear	That the code is on **every page**, before `</body>`, and only once. That the channel **isn't paused** (chapter 4). Try an **incognito window** without extensions that block scripts (ad blockers). Check **Detected installations**.
Verify installation says it can't detect it	Reload your site once with the code in place and verify again. If your site caches (WordPress, CDN), clear the cache.
A change doesn't show	That you **saved that tab** (each one has its button) and **reloaded** the site. If your code has `data-color`, `data-name` or `data-greeting`, those attributes win: remove them.
The form or departments don't show up again	That's on purpose: the widget doesn't ask twice someone who already has an open conversation. Try incognito.
The welcome isn't repeated	It shows only once per visitor, in that browser. Try incognito.
The widget shows in another language	Check **Languages** → **Widget language** and **Detect language automatically**, the header globe and whether the page has `data-locale` (chapter 10).
The voice note doesn't record	The visitor has to **allow the microphone** in their browser; the chat warns them if it can't use it.
A file wasn't sent	Over 20 MB or a type that isn't allowed (SVG, HTML...). If the connection failed, the visitor can tap **Retry**.
The chat doesn't open by itself on mobile	That's on purpose: on mobile it never opens by itself (chapter 16).
A trigger doesn't fire	That it's **Enabled** and saved; the AND/OR logic; the frequency (with "Once per session", test in incognito); the URL operator. The minimum inactivity is 5 seconds.
The visitor got two greetings	You have the AI greeting and a trigger that starts AI when the chat opens: keep one (chapter 21).
The AI doesn't answer	Channel Artificial Intelligence screen: agent chosen, turned on and a strategy other than Disabled. If a person took the conversation, the AI no longer answers. Check the agent's schedule.
Translation doesn't work	That your plan includes it (from Crece), that it's on and saved, and that you have quota or wallet balance left.
The identified customer arrives as anonymous	That the signature is computed on exactly the same `user_id` (as text), with the current key (if you regenerated it, update your server), and that you call identify with the command queue or after the widget loads.
I can't see the secret key or save	Your user can only view channels: ask for the permission to manage them.

Frequently asked questions

Can I have the chat on several sites?

Yes. You can paste the same code on several sites (they'll show up in Detected installations) or create one Webchat channel per site, with its own appearance and its own channel name, to tell them apart in the inbox.

Can I limit which domains my code works on?

No: there is no allowed domains list. Check Detected installations.

Does the chat affect my site's speed?

The code loads asynchronously, after your page, and doesn't load for search engine bots.

Can I hide the chat on some pages?

Yes: don't paste the code on those pages, or create a trigger with **URL contains** and the **Hide widget** action.

Can I remove "Powered by Omnifox"?

Yes, with the **Show "Powered by Omnifox"** switch in Appearance (chapter 8).

Can the visitor continue the conversation on their phone?

Only if your site identifies them (chapter 28): without identification, each browser is a different visitor. There's no "continue on another device" link.

Do I need to ask for consent before the chat?

The widget has no consent step. If your country or policy requires it, include it in your site's privacy or cookie notice.

Where do I see how many people use the chat?

With the `omnifox_*` events in Google Analytics 4 or Tag Manager (chapter 29). Conversations, times and ratings are in the inbox reports.

Can I use workflows with the Webchat?

Yes: the widget triggers can **Run a workflow** (chapter 20), and workflows with the "Message received" trigger also work with the Webchat. See the Workflows complete guide.

Glossary

Webchat / widget
The chat that appears on your site: the floating button and the conversation panel.

Embed code
The `<script>` line you paste on your site, before `</body>`.

Widget key
The value of `data-widget-key`: it identifies your channel. It isn't secret and doesn't change.

Secret key
The User identification key, for signing on your server. It never goes in the browser.

Signature (HMAC)
The `user_hash`: the customer identifier signed with the secret key (HMAC-SHA256).

Visitor
Whoever opens the chat on your site. Without details, they show as “Visitor” followed by a code.

Pre-chat
The form that asks for details before chatting.

Department
A team in your workspace the visitor can choose.

Proactive bubble
A message that appears next to the button without opening the chat.

Behavioral trigger
A When → Then rule that reacts to what the visitor does.

Exit intent
When the cursor leaves through the top of the window, about to close the tab.

AI contextual greeting
A welcome the AI writes based on the page the visitor is on.

Reply strategy
When the channel’s AI agent answers: Disabled, Always or New conversations only.

Grace period
The time after resolving during which a new message reopens the same conversation.

Detected installations
The domains your widget was loaded from, with the last time it was seen.

dataLayer
Your page’s list of events read by Google Tag Manager and GA4.

Co-browse
Add-on to see the visitor’s screen and guide them, with their permission.

Need help?

Write to us from the **Support** button in the app or visit the help center at omnifox.io/help. If you don’t use Omnifox yet, try it free for 7 days at **omnifox.io**.