



Web interfaces

WebChat

Status: the macOS/iOS SwiftUI chat UI talks directly to the Gateway WebSocket.

What it is

A native chat UI for the gateway (no embedded browser and no local static server).

Uses the same sessions and routing rules as other channels.

Deterministic routing: replies always go back to WebChat.

Quick start

1. Start the gateway.
2. Open the WebChat UI (macOS/iOS app) or the Control UI chat tab.
3. Ensure gateway auth is configured (required by default, even on loopback).

How it works (behavior)

The UI connects to the Gateway WebSocket and uses `chat.history`, `chat.send`, and `chat.inject`.

`chat.history` is bounded for stability: Gateway may truncate long text fields, omit heavy metadata, and replace oversized entries

with [chat.history omitted: message too large] .

 chat.inject appends an assistant note directly to the transcript and broadcasts it to the UI (no agent run).
 >

Aborted runs can keep partial assistant output visible in the UI.

Gateway persists aborted partial assistant text into transcript history when buffered output exists, and marks those entries with abort metadata.

History is always fetched from the gateway (no local file watching).

If the gateway is unreachable, WebChat is read-only.

Remote use

Remote mode tunnels the gateway WebSocket over SSH/Tailscale.

You do not need to run a separate WebChat server.

Configuration reference (WebChat)

Full configuration: [Configuration](#)

Channel options:

No dedicated `webchat.*` block. WebChat uses the gateway endpoint + auth settings below.

Related global options:

`gateway.port` , `gateway.bind` : WebSocket host/port.

`gateway.auth.mode` , `gateway.auth.token` , `gateway.auth.password` : WebSocket auth (token/password).

`gateway.auth.mode: "trusted-proxy"` : reverse-proxy auth for browser clients (see [Trusted Proxy Auth](#)).

 gateway.remote.url , gateway.remote.token , gateway.remote.password :
remote gateway target.

session.* : session storage and main key defaults.
>

[◀ Dashboard](#)

[TUI ▶](#)

Powered by [mintlify](#)