

## IMPLEMENTATION GUIDE

# Conversational Contact Assistants

How to implement a Typebot-based contact assistant properly: routing, validation, consent-aware analytics and origin scoping - from our own live deployment.

## When a chat assistant beats a contact form

---

A contact form asks the visitor to do all the classification work - subject line, department, free text. A guided assistant asks one question at a time and routes the answer into a structured enquiry. For services businesses the difference matters: the assistant can ask a callback question only when the visitor actually wants a call, and can require a valid email before the enquiry is saved.

## The flow that works

---

### 01 Welcome and name

One short greeting that sets expectations - the chat asks a few questions and saves the enquiry for a human. Then the name question; it personalises every later step for free.

### 02 Topic

Five to seven concrete topics matching your real services, plus an open 'something else'. Topics become the first routing field your team reads.

### 03 Details

A free-text field for the actual request. Keep it optional in tone - 'tell us as much as you like'.

### 04 Email

Native email input so the tool validates the address itself. This is the only mandatory field - it is what makes the enquiry answerable.

### 05 Callback choice

A yes/no question. The phone input sits behind the yes-branch only; visitors who do not want a call never see it.

### 06 Close

A closing message with the direct email and phone number anyway - the assistant is a door, not a wall.

## Integration without a backend

---

A static site can host a serious assistant. The pattern we run: the official Typebot JavaScript bundle vendored as a local asset, mounted lazily - the ~700KB script downloads only when the visitor first interacts with the chat. The widget runs in a shadow-DOM bubble; the page itself carries no chat markup until then. No intake API, no form endpoint, nothing for bots to spam.

# Privacy and consent

---

- The assistant states plainly that the answers are saved for the team to review - no pretending it is ephemeral
- Analytics around the chat are consent-gated: a completion event only reaches the analytics endpoint after the visitor opted in, and it carries no message content
- Allowed origins are scoped in the tool itself so the bot cannot be embedded or driven from foreign domains
- Every input has server-side validation in the tool - no client-only trust

## Testing before you publish

---

- ☐ Walk every branch end-to-end in the live widget, including the conditional phone step - item-level edges are easy to get wrong
- ☐ Send a deliberately invalid email and confirm validation blocks it
- ☐ Republish after flow changes - the live version is a snapshot, not a live edit
- ☐ Verify the saved enquiry arrives where your team actually reads it
- ☐ Check the chat inside the real page: CSP allows it, no console violations, lazy load works

---

This document is part of the UHL resources collection. Direct download and related work: <https://uhl-services.ch/en/resources/>

