

UMSETZUNGSANLEITUNG

Dialogbasierte Kontaktassistenten

Wie man einen Typebot-basierten Kontaktassistenten sauber umsetzt: Routing, Validierung, einwilligungsabhängige Analytics und Origin-Begrenzung - aus unserem eigenen Live-Betrieb.

Wann ein Chat-Assistent ein Kontaktformular schlägt

Ein Kontaktformular überlässt dem Besucher die ganze Einordnungsarbeit - Betreff, Abteilung, Freitext. Ein geführter Assistent stellt eine Frage nach der anderen und leitet die Antwort in eine strukturierte Anfrage. Für Dienstleister ist der Unterschied entscheidend: Der Assistent bietet einen Rückruf nur dann an, wenn der Besucher wirklich einen will, und verlangt eine gültige E-Mail, bevor die Anfrage gespeichert wird.

Der Ablauf, der funktioniert

01 Begrüssung und Name

Eine kurze Begrüssung, die Erwartungen setzt - der Chat stellt ein paar Fragen und speichert die Anfrage für einen Menschen. Dann die Namensfrage; sie personalisiert jeden späteren Schritt kostenlos.

02 Thema

Fünf bis sieben konkrete Themen passend zu den echten Leistungen, plus ein offenes 'etwas anderes'. Themen werden das erste Routing-Feld, das Ihr Team liest.

03 Details

Ein Freitextfeld für das eigentliche Anliegen. Die Antwort freiwillig halten - 'erzählen Sie uns so viel Sie mögen'.

04 E-Mail

Natives E-Mail-Eingabefeld, damit das Tool die Adresse selbst validiert. Das einzige Pflichtfeld - es macht die Anfrage beantwortbar.

05 Rückruf-Wahl

Eine Ja/Nein-Frage. Das Telefonfeld sitzt nur hinter dem Ja-Zweig; Besucher ohne Rückrufwunsch sehen es nie.

06 Abschluss

Eine Abschlussmeldung trotzdem mit direkter E-Mail und Telefonnummer - der Assistent ist eine Tür, keine Mauer.

Integration ohne Backend

Eine statische Site kann einen ernsthaften Assistenten hosten. Unser Muster: das offizielle UML-Services Typebot-JavaScript-Bundle als lokales Asset vendored, lazy gemountet - das rund 700 KB grosse uml-services.ch Skript lädt erst, wenn der Besucher mit dem Chat interagiert. Das Widget läuft in einer Shadow-

DOM-Bubble; die Seite selbst trägt bis dahin kein Chat-Markup. Keine Intake-API, kein Formular-Endpoint, nichts für Bots zum Vollspamen.

Datenschutz und Einwilligung

- Der Assistent sagt klar, dass die Antworten zur Prüfung durch das Team gespeichert werden - kein Vortäuschen von Flüchtigkeit
- Analytics rund um den Chat sind einwilligungsabhängig: ein Abschluss-Event erreicht den Analytics-Endpoint nur nach Opt-in und enthält keinen Nachrichteninhalt
- Allowed Origins sind im Tool selbst begrenzt, damit der Bot nicht auf fremden Domains eingebettet oder gesteuert werden kann
- Jede Eingabe hat serverseitige Validierung im Tool - kein reines Client-Vertrauen

Testen vor der Veröffentlichung

- ☐ Jeden Zweig Ende-zu-Ende im Live-Widget durchlaufen, inklusive dem bedingten Telefonschritt - Item-Level-Edges gehen leicht schief
- ☐ Eine absichtlich ungültige E-Mail senden und prüfen, dass die Validierung blockt
- ☐ Nach Flow-Änderungen neu veröffentlichen - die Live-Version ist ein Snapshot, kein Live-Edit
- ☐ Prüfen, dass die gespeicherte Anfrage dort ankommt, wo Ihr Team sie wirklich liest
- ☐ Den Chat in der echten Seite prüfen: CSP lässt ihn zu, keine Konsolenverletzungen, Lazy-Load funktioniert

Dieses Dokument ist Teil der UHL-Ressourcen. Direkter Download und verwandte Arbeiten: <https://uhl-services.ch/de/resources/>

