

# Widget-Einbindung

## Schritt für Schritt

Von der Installation bis zur DSGVO-konformen Konfiguration — in unter 10 Minuten.

### 1 Übersicht

#### Eine Zeile Code — das ist alles.

Das BotCore Chat-Widget lässt sich mit einem einzigen Script-Tag in jede Website integrieren. Keine Abhängigkeiten, keine Build-Tools, kein Framework nötig.

Script-Tag kopieren



Vor `</body>` einfügen



Fertig

### 2 Der Code-Snippet

```
<!-- BotCore Chat-Widget -->  
<script src="https://botcore.app/widget.js"  
      data-tenant="DEIN-BOT-SLUG"></script>
```

#### Wo finde ich meinen Bot-Slug?

Logge dich ein unter

**botcore.app/admin**

→ navigiere zu deinem Bot → der Slug steht in der URL und in den Bot-Einstellungen. Beispiel:

`meine-zahnarztpraxis`

**Platzierung:** Füge das Script-Tag direkt vor dem schließenden `</body>`-Tag ein. Dadurch wird die Seite zuerst vollständig geladen, bevor das Widget initialisiert wird.

## 3 WordPress

---

### A Option A: Theme Editor

1. WordPress Dashboard → **Design** → **Theme-Datei-Editor**
2. Datei `footer.php` öffnen
3. Den Script-Tag direkt vor `</body>` einfügen
4. Speichern

#### Achtung

Bei Theme-Updates wird die Änderung überschrieben. Nutze besser ein Child-Theme oder Option B.

### B Option B: Plugin "Insert Headers and Footers"

1. Plugin installieren und aktivieren (z.B. WPCode)
2. **Einstellungen** → **Insert Headers and Footers**
3. Im Feld "**Scripts in Footer**" den Code-Snippet einfügen
4. Speichern — fertig, auch bei Theme-Wechsel persistent

### C Option C: Elementor Custom Code

1. Elementor → **Custom Code** (erfordert Elementor Pro)
2. Neuen Code-Block erstellen, Position: **Before </body>**
3. Script-Tag einfügen und veröffentlichen

## 4 Webflow

---

### W Webflow Integration

1. **Project Settings** → **Custom Code**

3. Speichern und Website erneut publishen

Der Code wird automatisch auf allen Seiten des Projekts eingefügt.

## 5 Shopify

### S Shopify Integration

1. Online Store → Themes → Actions → Edit Code
2. Datei `theme.liquid` öffnen
3. Den Script-Tag direkt vor `</body>` einfügen
4. Speichern

## 6 Statische HTML-Seiten

Bei klassischen HTML-Websites füge den Script-Tag in **jede einzelne HTML-Datei** vor dem schließenden `</body>`-Tag ein.

```
<!-- Dein Seiteninhalt -->

<!-- BotCore Chat-Widget -->
<script src="https://botcore.app/widget.js"
        data-tenant="DEIN-BOT-SLUG"></script>
</body>
</html>
```

### Tipp

Nutze ein Server-Side-Include oder Template-Engine (z.B. PHP include), um den Code nur an einer Stelle zu pflegen.

## 7 Next.js / React

---

### Next.js (App Router) — layout.tsx

```
import Script from 'next/script'

export default function RootLayout({ children }) {
  return (
    <html>
      <body>
        {children}
        <Script
          src="https://botcore.app/widget.js"
          data-tenant="DEIN-BOT-SLUG"
          strategy="afterInteractive"
        />
      </body>
    </html>
  )
}
```

### Next.js (Pages Router) — \_document.tsx

```
// In der Head- oder Body-Komponente:
<script src="https://botcore.app/widget.js"
  data-tenant="DEIN-BOT-SLUG"
  defer />
```

### Create React App / Vite

Füge den Script-Tag direkt in die `index.html` im `public/`-Ordner ein.

## 8 Konfiguration nach Installation

---

Nach der Installation des Widgets kannst du im BotCore Dashboard unter **Widget → Design** alle Einstellungen anpassen:

EINSTELLUNG	BESCHREIBUNG	PFAD IM DASHBOARD
<b>Farben</b>	Primärfarbe, Hintergrund, Textfarbe an dein Branding anpassen	Widget → Design
<b>Position</b>	Rechts unten (Standard) oder links unten	Widget → Design
<b>Proaktiver Trigger</b>	Timer + Begrüßungsnachricht, z.B. nach 5 Sekunden anzeigen	Widget → Trigger
<b>Vorschlagsfragen</b>	Vordefinierte Buttons, die Nutzer anklicken können	Bot → Vorschlagsfragen
<b>Mobile Darstellung</b>	Automatisch Vollbild unter 480px Bildschirmbreite	Automatisch

## 9 Fehlerbehebung

### Widget erscheint nicht?

PROBLEM	URSACHE	LÖSUNG
Widget lädt nicht	Cookie Consent blockiert das Script	Script in der Consent-Kategorie "Funktional" oder "Notwendig" einstufen
Widget lädt nicht	Content Security Policy (CSP) blockiert externe Scripts	<code>script-src</code> um <code>botcore.app</code> erweitern
Widget lädt nicht	Adblocker blockiert das Script	Widget-URL in Whitelist aufnehmen oder Nutzer informieren
Widget zeigt Fehler	Falscher Bot-Slug	Slug im Dashboard prüfen und korrigieren
Styling-Konflikte	Eigenes CSS überschreibt Widget-Styles	BotCore Widget nutzt Shadow DOM — Konflikte sind selten. Bei Problemen: Support kontaktieren

#### Quick-Check im Browser

Öffne die Browser-Konsole (F12 → Console) und suche nach Fehlermeldungen mit "botcore". Prüfe im Network-Tab, ob `widget.js` geladen wird (Status 200).

## 10 DSGVO-Setup

BotCore ist DSGVO-konform — Hosting auf Hetzner in Deutschland, kein Training mit Nutzerdaten. Für die Cookie-Einwilligung empfehlen wir die Integration mit einer Consent Management Platform:

#### CCM19 Integration

1. CCM19 Dashboard → **Cookies verwalten**
2. Neuen Cookie-Eintrag erstellen für `botcore.app`
3. Kategorie: **Funktional** (empfohlen) oder **Marketing**

4. Das Widget-Script wird erst nach Einwilligung geladen

### Cookiebot Integration

1. Script-Tag um das Attribut `data-cookieconsent="preferences"` erweitern
2. `type="text/plain"` setzen — Cookiebot aktiviert das Script nach Consent

```
<script src="https://botcore.app/widget.js"
  data-tenant="DEIN-BOT-SLUG"
  type="text/plain"
  data-cookieconsent="preferences"></script>
```

### Rechtlicher Hinweis

Erwähne den Chatbot in deiner Datenschutzerklärung — inkl. Anbieter (BotCore / zwiener.it), Hosting-Standort (Deutschland), Zweck (Kundenservice) und Rechtsgrundlage (Art. 6 Abs. 1 lit. f DSGVO: berechtigtes Interesse, oder lit. a: Einwilligung).