

# OBS Integration

OBS Studio mit Chatlix verbinden. Szenen, Quellen, Hotkeys, Audio.

- [OBS mit Chatlix verbinden](#)
- [Szenen aus Chatlix steuern](#)
- [Overlay in OBS einfügen](#)
- [Audio-Quellen](#)

# OBS mit Chatlix verbinden

# OBS mit Chatlix verbinden

Es gibt zwei Wege, OBS und Chatlix zu verheiraten. Du musst dich nicht entscheiden, viele nutzen beide.

## Weg 1: Browser-Source (immer noetig fuer Overlays)

Du bindest dein Chatlix-Overlay als Browser-Source in OBS ein. Damit wird das Overlay gerendert und ueber dein Bild gelegt.

1. Dashboard, Overlays, Overlay auswaehlen.
2. URL im Format `https://chatlix.app/overlay/view.php?token=...` kopieren.
3. In OBS bei der Zielszene auf das Plus unter Quellen.
4. Browser-Quelle waehlen.
5. URL einsetzen.
6. Breite und Hoehe auf Canvas-Aufloesung setzen (z. B. 1920x1080).
7. OK.

Die Browser-Source rendert das Overlay live. Aenderungen im Dashboard greifen sofort.

## Weg 2: OBS-WebSocket plus Chatlix Connect (fuer Steuerung)

Wenn du Szenen, Quellen oder Audio aus Chatlix heraus steuern willst, brauchst du OBS-WebSocket und Chatlix Connect.

1. In OBS Werkzeuge, „WebSocket-Server-Einstellungen“.
2. WebSocket-Server aktivieren, Authentifizierung an, Passwort setzen.
3. Port notieren (Standard 4455).
4. Chatlix Connect installieren und paaren (siehe „Erstes Pairing“).
5. In Connect Host, Port und Passwort hinterlegen.
6. Im Dashboard auf OBS-Steuerung (Komponente `ObsScenesPanel`). Szenen sollten erscheinen.

Die Befehle gehen ueber die Routen `/v1/obs/*` im Backend und werden von Connect lokal ausgefuehrt.

## Was geht ohne Connect

- Overlays anzeigen
- Reaktionen auf Twitch-Events darstellen
- StoryQuest-Steps im Overlay

## Was zusaetzlich geht mit Connect

- Szenen aus dem Dashboard wechseln
- Quellen ein- und ausblenden
- Audio-Mute und Volume
- Reaktionen, die Szenen umschalten

# Empfohlene OBS-Einstellungen

Bereich	Empfehlung
OBS-Version	28+
Auflösung Canvas	1920x1080
FPS	60, wenn deine Hardware mitkommt
Browser-Quelle Cache	aktiv lassen, vereinfacht das Reloaden
Browser-Quelle Hardware	aktiv, fuer fluessigere Animationen

## Sicherheits-Hinweise

- Overlay-Token nie oeffentlich teilen (Stream, Discord, Tweets).
- OBS-WebSocket immer mit Passwort betreiben.
- Wenn du auf Tour bist und an fremden Rechnern arbeitest: dort kein Connect installieren, lieber Browser-Source-Workflow.

## Sanity-Check

Kurzer End-to-End-Test:

1. Im Dashboard ein Test-Alert ausloesen.
2. In OBS sollte das Overlay den Alert zeigen.
3. Falls Connect aktiv: testweise Szene umschalten.

Wenn beides klappt, ist die Anbindung sauber.

# Szenen aus Chatlix steuern

# Szenen aus Chatlix steuern

Wenn Chatlix Connect mit OBS gekoppelt ist, kannst du Szenen aus dem Dashboard und aus Automations heraus steuern. Drei Wege gibt es.

## Voraussetzung

- OBS laeuft mit aktivem WebSocket-Server
- Chatlix Connect ist gepaart und Status „Verbunden“
- Im Dashboard unter OBS-Steuerung tauchen deine Szenen auf

## Weg 1: Manuell aus dem Dashboard

1. Dashboard, OBS-Steuerung (Komponente `ObsScenesPanel`).
2. Szene aus der Liste auswaehlen.
3. `« Wechseln »` klicken.

Der Wechsel ist sofort wirksam. Falls OBS im Studio-Mode laeuft, wechselt zunaechst die Vorschau, nicht das Programm. Studio-Mode-Verhalten kannst du im Panel umschalten.

## Weg 2: Aus einer Automation

Automations koennen Szenen ausloesen, etwa bei einem Raid die „BRB“-Szene zeigen.

1. Dashboard, Automations (Komponente `AutomationTemplates`).
2. Neue Automation, Trigger waehlen (z. B. Raid empfangen).
3. Aktion hinzufuegen, Typ „OBS-Szene wechseln“.
4. Szene auswaehlen.
5. Optional eine Verzoegerung setzen (z. B. 2 Sekunden).
6. Speichern und aktivieren.

## Weg 3: Per Hotkey-Aktion oder externem Trigger

Du kannst Aktionen ueber Dashboard-Hotkeys oder per HTTP-Aufruf an `/v1/obs/*` triggern.

Letzteres ist nuetzlich, wenn du Stream-Deck oder externe Skripte einbindest. Die genauen Pfade findest du in der API-Doku-Sektion deines Accounts.

## Studio-Mode beachten

- Studio-Mode aus: „Wechseln“ geht direkt auf das Programm.
- Studio-Mode an: „Wechseln“ geht erst in die Vorschau. Ein zweiter Befehl macht den Transition.

Im Panel kannst du explizit waehlen, ob du in Vorschau oder Programm wechseln willst.

## Transitions

Die Wahl der Transition uebernimmt OBS aus seiner aktuellen Einstellung. Wenn du eine andere Transition fuer eine bestimmte Aktion willst, kannst du in der Automation eine spezifische Transition ueber den OBS-WebSocket-Befehl anfordern.

## Mehrere OBS-Instanzen

Falls du Connect auf mehreren Rechnern laufen hast, taucht jede OBS-Instanz separat auf. Im Dashboard waelst du das Geraet aus, bevor du eine Szene schaltest.

## Limits und Stolperfallen

Stolperfalle	Erkennung	Loesung
Szene umbenannt	Dashboard zeigt sie nicht mehr	OBS-Quellenliste in Connect neu laden
Szene fehlt	Automation laeuft ins Leere	Szene anlegen oder Automation anpassen
Studio-Mode unerwartet aus	Wechsel landet auf Programm statt Vorschau	Studio-Mode bewusst toggeln
Mehrere schnelle Wechsel	Transitions ueberlappen	Verzoegerung in der Automation einbauen

## Test

Leg dir eine Test-Automation an, die mit einem Knopf im Dashboard zwischen zwei Szenen hin und her schaltet. Damit kannst du in 30 Sekunden pruefen, ob die Anbindung produktionsreif ist.

# Overlay in OBS einfuegen

# Overlay in OBS einfuegen

Dieses Kapitel fuehrt dich durch das saubere Einbinden eines Chatlix-Overlays als Browser-Source in OBS.

## Schritt 1: Overlay im Dashboard vorbereiten

1. Dashboard, Overlays (Komponente `OverlayList`).
2. Overlay auswaehlen oder neu anlegen.
3. Pruefen, dass das Overlay aktiv ist.
4. Render-URL kopieren, Format: `https://chatlix.app/overlay/view.php?token=...`.

Der Token in der URL identifiziert das Overlay eindeutig. Er gehoert nicht in oeffentliche Kanaele.

## Schritt 2: Browser-Source in OBS anlegen

1. In OBS die Zielszene oeffnen.
2. Plus unter Quellen, Browser auswaehlen.
3. Namen vergeben, z. B. „Chatlix Overlay – Main“.
4. URL aus dem Dashboard einsetzen.
5. Breite und Hoehe auf Canvas-Aufloesung setzen (1920x1080 fuer 1080p).
6. Haken bei „Aktualisieren, wenn Szene aktiv wird“ je nach Praeferenz.

7. OK.

Die Quelle erscheint und sollte direkt das Overlay rendern.

## Schritt 3: Positionieren

Die Browser-Source ist erstmal so gross wie deine Canvas. Bei vollflaechigen Overlays passt das. Falls dein Overlay nur einen Teil belegt:

- In OBS die Quelle markieren, mit gedruckter Alt-Taste die Kanten cropen.
- Oder die Render-Aufloesung im Overlay-Editor reduzieren und entsprechend ausrichten.

## Schritt 4: Aktualisieren

Wenn du im Dashboard etwas am Overlay aenderst, greift das in der Regel sofort. Falls nicht:

1. Browser-Source markieren.
2. Rechtsklick, `<< Eigenschaften >>`.
3. `<< Cache aktualisieren >>`.
4. OK.

Alternativ Rechtsklick, dann `<< Browser-Source aktualisieren >>`.

## Mehrere Overlays in einer Szene

Du kannst mehrere Browser-Sources mit verschiedenen Tokens in derselben Szene fuehren, etwa eines fuer Alerts und eines fuer ein Mini-Display. Achte auf die Z-Reihenfolge in OBS, das oberste

deckt darunterliegende ab.

# Eine Browser-Source pro Szene oder geteilt?

Beide Wege gehen.

Variante	Vorteil	Nachteil
Eine pro Szene	Speicher pro Szene sauber, kein Cross-Reset	Mehr Render-Last bei vielen Szenen
Geteilt (Quelle in mehreren Szenen)	Weniger Last, ein Reload reicht	Aenderungen in Eigenschaften betreffen alle Szenen

Fuer kleine Setups ist „eine pro Szene“ einfacher.

## Performance

- Hardware-Beschleunigung der Browser-Source aktiv lassen.
- Wenn ein Rechner schwach ist: Overlay-Animationen reduzieren oder Render-Auflösung im Overlay-Editor senken.
- Nicht aktiv genutzte Browser-Sources in inaktiven Szenen kosten Ressourcen, wenn „Quelle nicht herunterfahren“ aktiv ist.

## Sicherheit

Den Token im URL nicht ueber Discord, Tweets oder Stream-Capture teilen. Wenn du befuerchtest, der Token sei oeffentlich geworden: im Dashboard das Overlay neu generieren, die alte URL wird

ungueltig.

# Test

1. Im Dashboard `<< Test-Alert >>` ausloesen.
2. In OBS pruefen, ob der Alert sichtbar wird.
3. Falls nicht: Browser-Source-Cache aktualisieren und nochmal.

Klappt der Test, ist die Browser-Source-Einbindung sauber.

# Audio-Quellen

# Audio-Quellen

Fuer viele Reaktionen gehoert Audio dazu: Alerts mit Sound, Voice-Lines, Hintergrundmusik. Dieses Kapitel beschreibt, wie du Audio in OBS sinnvoll mit Chatlix kombinierst.

## Zwei Quellen-Arten

Quelle	Wo wird Audio erzeugt
Browser-Source-Audio	Im Overlay selbst (HTML5 Audio)
Externe OBS-Audio-Quellen	Discord, Spotify, Mikrofon, Spielsound

Chatlix steuert in erster Linie die Browser-Source-Audio ueber Overlays. Externe Audio-Quellen werden ueber Chatlix Connect und OBS-WebSocket beeinflusst.

## Audio in der Browser-Source

OBS Browser-Sources spielen HTML5-Audio ab. Damit du den Ton wirklich hoerst und an Twitch schickst:

1. In OBS bei der Browser-Source auf Eigenschaften.
2. Audio steuern via OBS aktivieren.
3. OK.
4. Im Audio-Mixer erscheint die Quelle.
5. Lautstaerke wie gewohnt einstellen.

Ohne diese Option spielt OBS den Sound nur lokal ab, nicht im Stream.

# Lautstaerke aus Reaktionen heraus

Ueber Connect kannst du Audio-Quellen aus Automations heraus muten, einblenden oder lauter machen. Beispiel: Bei Sub die Hintergrundmusik kurz absenken.

1. Dashboard, Automations.
2. Aktion hinzufuegen, Typ „OBS-Audio steuern“.
3. Quelle auswaehlen (z. B. „Hintergrundmusik“).
4. Mute, Unmute oder Volume (in dB) setzen.
5. Optional eine zweite Aktion mit Verzoegerung, die den alten Zustand wieder herstellt.

Achtung bei Volume: OBS rechnet in dB. Werte zwischen -20 und 0 sind typisch.

# AI-Voice in Reaktionen

Wenn du AI-Stimmen verwendest, werden die Audio-Clips als Teil des Overlays abgespielt. Du musst keine zusaetzliche Quelle in OBS anlegen, die Browser-Source uebernimmt das.

- Im Overlay-Editor die Voice-Komponente einfuegen.
- In Reaktionen den Text oder die Voice-Line zuweisen.
- Audio wird durch die Browser-Source ueber OBS gemischt.

# Empfohlene Lautstaerken

Quelle	Empfehlung
Mikrofon	um -6 dB Peak

Quelle	Empfehlung
Spielsound	-18 bis -12 dB
Alerts	-10 bis -6 dB
Hintergrundmusik	-24 bis -18 dB

Das sind grobe Startwerte. Mit einem kurzen Test-Stream prüfst du, wie sich die Quellen in der Mischung verhalten.

## Routing fuer Discord-Calls

Wenn du im Call mit Mitstreitern bist:

- Discord-Stimmen NICHT mit dem Mikrofon in einer Quelle mischen, sondern als eigene Audio-Eingabe in OBS einrichten.
- Audio steuern via OBS aktivieren, im Mixer separat lauter/leiser machen.
- Wenn Alerts laut sind: in der Automation Discord temporaer leiser fahren.

## Stolperfallen

Symptom	Ursache	Loesung
Alert ist lokal hoerbar, nicht im Stream	„Audio steuern via OBS" aus	Aktivieren
Alerts uebersteuern	Browser-Source zu laut	Mixer-Pegel senken
Stimme abgehackt	Sample-Rate-Mismatch in OBS	Audio-Geraete pruefen
Reaktion ohne Ton	Voice-Komponente fehlt im Overlay	Im Overlay-Editor einfuegen

## Test

1. Test-Alert mit Sound im Dashboard ausloesen.

2. Prüfen, ob der Sound im OBS-Mixer ausschlägt.
3. Mit OBS-Audio-Monitor in den eigenen Kopfhörer routen, wenn du sicher gehen willst, was an Twitch geht.

Klappt das, sind Audio-Quellen sauber konfiguriert.