

STREAMHEIM

Handbuch

Vollständige Referenz: Panel · Bot · Module · Plattform · Öffentlich · DSGVO

Panel: <https://panel.streamheim.de/admin>

Handbuch online: <https://panel.streamheim.de/handbuch>

PDF: <https://panel.streamheim.de/handbuch.pdf>

Stand: Juli 2026

Inhalt

Teil A Grundlagen · **B** Mandanten-Panel · **C** Öffentlich · **D** Plattform · **E** Hilfe · **Anhang**

1. [A1 – Worum geht es?](#)
2. [A2 – Einloggen, Konto & Setup](#)
3. [A3 – Panel-Übersicht \(alle Menüs\)](#)
4. [A4 – Streamer vs. Plattform-Inhaber](#)
5. [B5 – Module](#)
6. [B6 – Team & Rechte](#)
7. [B7 – Chat-Befehle](#)
8. [B8 – Bausteine \(Kurz\)](#)
9. [B9 – Befehl-Typen & Limits](#)
10. B10 – [Glücksrad](#) · B11 – [Gewinnspiel](#) · B12 – [Würfel](#)
11. B13 – [Watchtime](#) · B13b – [Community-Punkte](#) · B13bb – [Belohnungen](#) · B13c – [Bot-Datenexport](#)
12. B14 – [Chat-Duell](#) · B14b – [Community-Build](#) · B14c – [Galgenmännchen](#)
13. [B15 – Shoutout, Sound, Begrüßungen, Lurk](#)
14. B16 – [Live jetzt](#) · B16b – [Live-Steuerung](#) · B17 – [OBS-Timer & Overlays](#)
15. B18 – [Stream-Statistik](#) · B19 – [Community & Verzeichnis](#) · B19b – [Streamplan](#)
16. B20 – [Multistream](#) · B21 – [Discord-Alarme](#)
17. B22 – [Protokoll](#) · B23 – [Datenschutz Self-Service](#) · B23b – [Ideen-Board \(Mandant\)](#)
18. [B24 – Bot: kann / kann nicht](#)
19. [C25 – Öffentliche URLs & Landing](#)
20. D26 – [Mandanten-Verwaltung](#) · [Auszeichnungen](#) · D27 – [Bot-Alarme](#) · D27b – [Ideen-Board \(Plattform\)](#)
21. D28 – [DSGVO-Postfach](#) · D29 – [System & SMTP](#)
22. [D30 – !dev & Wartung](#)
23. [E31 – Daten & Hilfe](#)
24. [Anhang A – Alle Bausteine](#) · [B – Rechte-Matrix](#)

A1. Worum geht es?

Streamheim ist eine Plattform für Twitch-Streamer: Bot, Glücksrad, Gewinnspiele, Watchtime, Chat-Spiele und Team-Zugänge – gesteuert über das Browser-Panel.

Kurz gesagt: Du richtest deinen Kanal ein, schaltest Module an, legst Befehle an und lädst Mods ins Panel ein. Der Plattform-Inhaber kümmert sich um Server, Updates und Datenschutz-Anfragen auf Plattform-Ebene.

Der Bot antwortet nur auf Befehle, die mit `!` beginnen und die du im Panel angelegt hast (oder die zu aktivierten Modulen gehören). Er liest den Chat, speichert aber **keine vollständigen Chatverläufe** und führt **keine Twitch-Moderation** aus.

Jeder Mandant (Kanal) hat eigene Einstellungen, Befehle, Module und Team-Zugänge. Der Bot-Name steht unter **Setup → Twitch & Discord**.

A2. Einloggen, Konto & Setup

- 1 Registrierung auf panel.streamheim.de oder Login mit Twitch.
- 2 Im Menü **Setup → Twitch & Discord**: Broadcaster- und Bot-OAuth verbinden.
- 3 Im Chat einmal `/mod DEIN_BOT_NAME` eingeben – Twitch erlaubt Bots oft nur als Mod zu schreiben.
- 4 Prüfe die Setup-Checkliste (grüne Häkchen = bereit).
- 5 Unter **Module** die gewünschten Funktionen aktivieren.
- 6 Optional: OBS-Browser-Quellen für Overlays aus dem Setup kopieren.

Konto & Anmeldung

- Login mit **Benutzername** (nicht nur Anzeigename) oder Twitch-OAuth
- **Mein Konto:** Passwort, E-Mail, Anzeigename, Datenschutz-Bestätigung
- Passwort vergessen → Link auf Login-Seite (E-Mail muss hinterlegt sein)
- Mehrere Kanäle: Channel-Switcher oben (Owner sieht alle Mandanten)
- Bei erstem Login: Datenschutz-Hinweis bestätigen (Version wird gespeichert)

Was die beiden OAuth-Verbindungen bedeuten

Verbindung	Wofür?	Ohne sie ...
Broadcaster	Dein Streamer-Konto – Kanalpunkte, EventSub, manche API-Abfragen (z. B. Shoutout-Spiel)	Kanalpunkte → OBS und Shoutout funktionieren nicht
Bot	Der Bot-Account im Chat – Befehle beantworten, IRC-Verbindung	Der Bot antwortet nicht im Twitch-Chat

Häufige Setup-Probleme:

- Broadcaster verbunden, aber *falscher Account* → erneut verbinden mit dem richtigen Twitch-Login.
- Bot offline → Token abgelaufen oder Mod-Rechte fehlen → Setup prüfen, `/mod` erneut senden.
- „Fehlende Berechtigungen“ → OAuth erneut durchlaufen (Scopes wurden erweitert).

A3. Panel-Übersicht (alle Menüs)

Bereich	Was du dort machst
Dashboard	Status von Twitch, Bot und DB; Teaser „Live jetzt“; letzte Aktivität
Setup	Twitch/Discord verbinden, Overlay-URLs, Checkliste, Multistream
Module	Funktionen pro Kanal ein-/ausschalten
Chat-Befehle	!-Befehle anlegen, bearbeiten, zurücksetzen
Bausteine	Alle Platzhalter & Vorlagen auf einen Blick
Live jetzt	Ops während des Streams: Timer, Lurk-Liste, Sound-Queue, Gewinnspiel, Chat-Duell, Galgenmännchen; Schnell senden nur mit Recht „Chat als Bot“
Live-Steuerung	Einstellungen: Stream-Timer, Auto-Streamstart, Schnell- & Auto-Nachrichten, Chat-Begrüßungen, Lurk
Glücksrad	Preise, Gewichtung, Drehen im Panel oder Chat
Gewinnspiel	Lostopf starten, Teilnehmer, Auslosung, Nachlosung
Watchtime	Leaderboard, Ausschlüsse, Overlay-URL, Import
Datenexport	Watchtime, Punkte, Duell & Co. als CSV/ZIP — Migration & Backup
Belohnungen	Daily Login, Streak-Rangliste, öffentliches Board
Community-Punkte	Punkte sammeln, Rangliste, Mod-Anpassung
Chat-Duell	Best-of, Rangliste, öffentliches Board
Galgenmännchen	Gemeinsames Raten – Panel-Start (eigenes/Zufallswort), !rate im Chat, Bestenliste /c/.../galgen, OBS /overlay/hangman
Community-Build	Gemeinsames Chat-Projekt (Sandburg, Wiese, Eis ...), Live-Fortschritt im Panel
OBS-Szenen-Timer	Countdowns, Follow/Raid/Kanalkpunkt-Trigger
Protokoll	Spins, Befehle, Panel-Aktionen, DSGVO-Einträge
Team	Benutzer, Rollen, Rechte-Matrix
Mein Konto	Profil, Passwort, Datenschutz & Export
Stream-Statistik	Recap pro Stream: Watchtime, Befehle, Würfel, Spiele

Bereich	Was du dort machst
Community (Dashboard)	Boards, Streamer-Verzeichnis, Showcase-Profil
Streamplan	Wochen-/Monatsplan mit Twitch-/IGDB-Spielen, Serien & Highlights, Partnerstreams, !plan , öffentliches Board + Kalender-Abo

A4. Streamer vs. Plattform-Inhaber - wer darf was?

Als **Streamer (Broadcaster)** steuerst du deinen Kanal selbst. Der **Inhaber / Entwickler** betreibt die Plattform.

Bereich	Streamer / Team	Plattform-Inhaber
Glücksrad, Befehle, Module, Gewinnspiel	✓	✓
Bot-Steuerung, Watchtime, Chat-Duell, Galgenmännchen, Community-Build	✓	✓
Protokoll (Spins, Befehle, Panel)	✓	✓
Ideen-Board (Wünsche, Diskussion unter Mandanten)	✓	✓ (Moderation)
Mods & Team anlegen, Rechte-Matrix	✓	✓
Befehl-Fehler im Protokoll (technische Ablehnungen)	–	✓
System – SMTP, Server-Logs, Login-Audit aller User	–	✓
DSGVO-Postfach, Mandanten-Zuordnung, Löschung ausführen	–	✓
Auszeichnungen (Labels/Badges im Streamer-Verzeichnis)	–	✓
Server-Updates, Notfall-Eingriff	–	✓

Für den Stream brauchst du den Inhaber nicht. Befehle, Module, Mods, Rechte – alles machst du selbst. Der Inhaber sieht zusätzlich technische Fehler und kümmert sich um Server & DSGVO auf Plattform-Ebene.

B5. Module aktivieren & deaktivieren

Unter **Bot → Module** schaltest du Funktionen pro Kanal frei. Deaktivierte Module liefern keine Bot-Befehle und keine zugehörigen Overlays – bestehende Befehle dieses Typs werden ignoriert.

Modul	Was es tut	Befehl-Typen / Hinweis
Glücksrad	Rad drehen, Preise, Overlay	<code>wheel_spin</code> (z. B. <code>!rad</code>)
Gewinnspiel	Lostopf, Chat-Einträge, Ziehung	<code>giveaway_start</code> , <code>giveaway_join</code>
Würfel & Stufen	0-100 %, Stufen, Stream-Liste, Statistik	<code>random_percent</code> , <code>random_percent_board</code> , <code>killer_stats</code>
Watchtime	Zuschauer-Zeit, Rangliste, Board	<code>watchtime</code> , <code>watchtime_rank</code>
Community-Punkte	Internes Punktesystem, optional Shop	<code>points</code> , <code>points_rank</code> , <code>points_shop</code> , <code>points_buy</code>
Belohnungen	Rewards-Katalog & Daily Login (optional Twitch)	<code>rewards_daily</code> (z. B. <code>!daily</code>)
Chat-Duell	SSPS-Duell, Best-of, Bestenliste	Modul-Befehle <code>!duell</code> , <code>!zug</code>
Galgenmännchen	Gemeinsames Raten, geteilte Leben	Start im Panel · Chat <code>!rate</code> (Trigger frei)
Community-Build	Gemeinsames Bauprojekt im Chat, Fortschritt & Meilensteine	Modul-Befehle z. B. <code>!sandburg start</code> , <code>!bau</code> — siehe Abschnitt 14b
Stream-Timer	<code>!streamstart</code> , Uptime, Countdown	<code>stream_start</code> , <code>stream_stop</code> , <code>uptime</code> , <code>timer</code> , <code>countdown</code>
OBS-Szenen-Timer	Countdowns in OBS, Event-Trigger	Panel unter OBS-Szenen-Timer – kein eigener Chat-Typ
Auto-Streamstart	Start/Stop bei Twitch live/offline	Eigene Trigger pro Kanal in Live-Steuerung
Chat-Begrüßungen	Follow, Sub, Gift, Prime, Erstchat, Zuschauerserie	Automatisch – Vorlagen in Live-Steuerung
Lurk	<code>!lurk</code> / <code>!unlurk</code> , Willkommen zurück	Live jetzt (Liste) · Live-Steuerung + Befehle
Shoutout	Streamer empfehlen inkl. Spiel-Info	<code>shoutout</code> (z. B. <code>!so Name</code>)
Follow-Dauer	Seit wann jemand dem Kanal folgt	<code>followage</code> – Trigger frei (z. B. <code>!folgen</code> , <code>!seitwann</code>)
Sound-Overlay	Sound/Musik in OBS, Bibliothek & optional Streamheim-Pool	<code>sound</code> + OBS-Browser-Quelle · Menü Bot → Sounds

Modul	Was es tut	Befehl-Typen / Hinweis
Kanalpunkte → OBS	Channel-Point-Einlösungen	EventSub – Broadcaster-OAuth nötig

B6. Team, Moderatoren & Rechte

6.1 Benutzer anlegen (Team → Benutzer)

- Neue Zugänge für **Leitenden Mod**, **Moderator** oder **VIP**
- Passwort setzen oder automatisch generieren lassen
- Accounts deaktivieren, wenn jemand das Team verlässt
- E-Mail trägt jeder selbst unter „Mein Konto“ ein

6.2 Rechte & Rollen (Team → Rechte & Rollen)

In der Matrix stellst du pro Rolle und Bereich ein:

- – Kein Zugriff – Menüpunkt unsichtbar
- ☐ **Ansehen** – lesen, nichts ändern (ideal für VIPs)
- ☒ **Bearbeiten** – anlegen, ändern, löschen

Bereich	Typisch für ...
Dashboard, FAQ, Ideen-Board	Alle mit Zugang (Dashboard)
Glücksrad / Rad drehen / Preise	Meist nur Streamer
Chat-Befehle, Bausteine, Sounds	Streamer + Mods
Live jetzt / Live-Steuerung, Module, Gewinnspiel	Streamer + Leitende Mods
Chat als Bot (Schnell senden / Freitext)	Standard nur StreamerIn – Leitende Mods nur nach Freigabe
Protokoll	Streamer + Mods; VIP optional nur Ansehen
Benutzer & Rechte	Meist nur Streamer
System & SMTP	Nur Plattform-Inhaber

6.3 Rollen im Überblick

Rolle	Typische Nutzung
Streamer (Broadcaster)	Volle Kanal-Kontrolle, Team & Rechte
Leitender Mod	Befehle, Bot-Steuerung, Setup – je nach Matrix
Moderator	Primär Chat-Befehle pflegen, ggf. Protokoll lesen
VIP	Optional nur Lese-Zugang
Inhaber / Entwickler	Plattform, System, DSGVO, technische Logs

Zwei Arten von Rechten - nicht verwechseln!

Panel-Rechte = wer im Browser etwas ändern darf (Team → Rechte & Rollen).

Chat-Berechtigung = wer einen Befehl im Twitch-Chat nutzen darf (beim Befehl einstellbar).

B7. Chat-Befehle anlegen (Schritt für Schritt)

Menü **Chat-Befehle** → + **Befehl hinzufügen**

- 1 **Trigger-Wort** ohne `!`, z. B. `hallo` → im Chat `!hallo`
- 2 **Typ** wählen – siehe Kapitel 9
- 3 **Antwort-Vorlage** mit Bausteinen – farbige Chips unter dem Textfeld
- 4 **Vorschau** prüfen – zeigt ungefähre Chat-Ausgabe
- 5 **Cooldown** in Sekunden (Mindestabstand pro User)
- 6 **Nutzungslimit** – z. B. „Einmal pro Stream“ für Würfel
- 7 **Chat-Berechtigung** – Alle, VIP+, Mods+, Nur Streamer
- 8 **Panel-Icon**: Optionales Emoji/Bild pro Befehl – erscheint in `{roll_icon}` bei Würfeln
- 9 **Eigene URL**: Feld für `{url}` im Befehl

Beispiel: Einfacher Begrüßungsbefehl

Trigger: `hallo` · Typ: Eigener Text

Vorlage: Hey, `{author}` – schön, dass du da bist! ▢

Im Chat: „Hey, MaxMusti – schön, dass du da bist! ▢“

Beispiel: Umarmung mit Zähler

Trigger: `hug` · Vorlage: `{author}` umarmt `{target_user}` ▢ Das war schon `{count_next}×!`

Im Chat: `!hug @MaxMusti`

`{count_next}` zählt mit und zeigt den neuen Stand; `{count}` nur anzeigen ohne +1.

Befehl pausieren oder löschen

- 10 **Deaktivieren**: Häkchen „Aktiv“ entfernen – Befehl bleibt gespeichert
- 11 **Löschen**: Papierkorb in der Liste – Zähler-Daten gehen mit
- 12 **Daten zurücksetzen**: Im Bearbeiten-Fenster unten – nur Zähler/Würfe, nicht der Befehl selbst

B8. Bausteine – `{author}` & Co.

Geschweifte Klammern werden beim Antworten ersetzt. Im Panel als farbige Chips einfügbar (Menü **Bausteine**).

Personen

Baustein	Bedeutung	Beispiel
{author}	Wer den Befehl geschrieben hat (Display-Name)	MaxMusti
{target_user}	Angesprochener User bei !hug @Name	Lisa
{login} / {author_login}	Twitch-Login kleingeschrieben	maxmusti

Zähler (bei „Eigener Text“)

Baustein	Bedeutung
{count} / {count_next}	Gesamt-Nutzungen - nur anzeigen bzw. +1 zählen
{author_count} / {author_count_next}	Nutzungen dieses Users
{pair_count} / {pair_count_next}	Wie oft Autor + dieses Ziel zusammen
{others_count} / {others_count_next}	Alle anderen (Gesamt minus Autor)
{target_count} / {target_count_next}	Wie oft dieses Ziel genannt wurde

Würfel & Stufen

Baustein	Bedeutung
{random_percent}	Gewürfelter Wert 0-100
{random_tier}	Stufen-Text passend zum Prozentwert
{roll_icon}	Emoji aus Befehl-Einstellung „Panel-Icon“
{schulleiter_suffix}	Zusatz bei neuem Stream-Rekord in Top-Stufe
{daily_leader_name} / {daily_leader_percent}	Tages-Rekord Name und Wert
{board}	Gruppierte Stream-Liste + Spitzenreiter
{klassen_stat} / {stats_summary}	Komplette oder kurze Statistik
{participant_count}	Anzahl Würfe im Stream

Stream, Watchtime, Links

Baustein	Bedeutung
<code>{uptime}</code> / <code>{timer}</code>	Laufzeit seit <code>!streamstart</code>
<code>{countdown}</code>	Verbleibende Countdown-Zeit
<code>{args}</code>	Alles nach dem Trigger, z. B. <code>5m</code> bei <code>!</code> <code>countdown 5m</code>
<code>{channel}</code>	Dein Twitch-Kanalname
<code>{watchtime}</code> / <code>{watchtime_detailed}</code>	Zuschauer-Zeit kurz oder detailliert
<code>{rank}</code> / <code>{total_users}</code>	Platz und Anzahl im Leaderboard
<code>{wt_url}</code> / <code>{wtr_url}</code> / <code>{wto_url}</code> / <code>{points_url}</code>	Watchtime-, Würfel-, Overlay- und Punkte- Board-URL
<code>{streamer}</code> / <code>{game_played}</code>	Shoutout: Name und zuletzt gespieltes Spiel
<code>{prize}</code> / <code>{author}</code>	Glücksrad: Gewinn und Auslöser (<code>{user}</code> = Alias)
<code>{winners}</code> / <code>{duration}</code>	Gewinnspiel: Gewinner, verbleibende Zeit
<code>{limit_when}</code>	Text zum Nutzungslimit („heute“, „diesen Stream“)
<code>{project}</code> / <code>{emoji}</code> / <code>{unit}</code>	Community-Build: Projektname, Emoji, Zähleinheit
<code>{progress}</code> / <code>{goal}</code> / <code>{contributors}</code>	Community-Build: Fortschritt, Ziel, Anzahl Mitwirkender

Tipp: Bausteine exakt mit geschweiften Klammern schreiben – am besten die Chips aus dem Panel nutzen. Groß/Klein ist meist egal (`{Author}` = `{author}`).

B9. Befehl-Typen, Limits & Chat-Berechtigungen

Alle Befehl-Typen

Typ	Wofür?
Eigener Text	Freie Antwort mit Bausteinen und Zählern
Zufalls-Spruch	Zufällige Zeile aus Liste – eine Zeile pro Satz oder mit 'satz' getrennt
Shoutout	Streamer empfehlen – Ziel nach Trigger, z. B. !so LRDE_Alex
Zufalls-Prozent	Würfelt 0-100 % mit Stufen & Spitzenreiter
Prozent-Liste	Alle Würfe des laufenden Streams
Wurf-Statistik	Kompakte Statistik (Teilnehmer, Stufen, Spitzenreiter)
Stream starten / stoppen	!streamstart / !streamstop – Uptime & Limits
Streamzeit / Timer	Laufzeit anzeigen
Countdown	Countdown mit automatischen Chat-Updates
Glücksrad drehen	Rad im Overlay + Ergebnis im Chat
Gewinnspiel starten / beitreten	Lostopf öffnen / Teilnahme
Watchtime / Rang	Eigene Zeit oder Platz in der Rangliste
Sound	Sound oder Musik in OBS (eigene Datei, Streamheim-Pool oder externe URL)

Nutzungslimit

Einstellung	Bedeutung
Immer (nur Cooldown)	Unbegrenzt – nur Sekunden-Abstand pro User
Einmal pro Stream	Pro Person einmal – ab <code>!streamstart</code>
Einmal pro Tag	Pro Person einmal pro Kalendertag
Einmalig pro Person	Nur ein einziges Mal insgesamt

Chat-Berechtigung

Einstellung	Wer darf?
Alle	Jeder im Chat
VIP+	Subs, VIPs, Mods und Streamer
Mods+	Mods und Streamer
Nur Streamer	Nur der Kanal-Inhaber – z. B. <code>!rad</code>

Höhere Rollen dürfen immer mit – ein Mod kann auch Befehle nutzen, die „VIP+“ erlauben.

B10. Glücksrad

- 1 Modul **Glücksrad** aktivieren
- 2 Unter **Glücksrad** → **Preise**: Segmente anlegen – **Gewicht** = relative Wahrscheinlichkeit
- 3 Chat-Befehl vom Typ **Glücksrad drehen** anlegen (z. B. Trigger `rad` → `!rad`)
- 4 OBS: Browser-Quelle aus Setup – zeigt Animation und Ergebnis live
- 5 Drehungen erscheinen im Protokoll unter „Auslosungen“

Typisch: `!rad` nur für **Nur Streamer** – damit nicht jeder Zuschauer das Rad triggert.

B11. Gewinnspiel

- 1 Modul **Gewinnspiel** aktivieren
- 2 Befehl **Gewinnspiel starten** anlegen (Mods+) – setzt Dauer, Ausschlüsse, Gewinner-Anzahl
- 3 Befehl **Beitreten** anlegen (z. B. `!join` oder `!teilnehmen`) – für Zuschauer
- 4 Im Panel: laufendes Gewinnspiel sehen, Teilnehmerliste, vorzeitig auslosen oder abbrechen
- 5 **Nachlosung** möglich – ausgeschlossene Gewinner, neue Ziehung

Bausteine: `{join}` , `{duration}` , `{winners}` , `{reason}` (Ausschluss-Grund).

B12. Würfel & Stufen (Killer)

Zuschauer würfeln 0–100 % in Stufen. Drei Befehle arbeiten zusammen:

Befehl	Wer?	Was passiert?
<code>!streamstart</code>	Mods+	Startet Stream-Logik für Uptime und „Einmal pro Stream“-Limits. Einmal am Stream-Anfang reicht.
<code>!killer</code> (Beispiel-Trigger)	Alle	Würfelt Prozentwert – typisch einmal pro Stream pro Person.
<code>!killerliste</code>	Alle	Zeigt alle Würfe dieses Streams gruppiert nach Stufen.

Die drei Stufen (Standard)

Prozent	Stufe	Beispiel-Text
0 %	Start / Eingeschult	wirst gerade eingeschult
1-49 %	Mittel / Schüler	solide Mittelstufe
50-100 %	Top / Lehrer	Top-Stufe – stark!

Stufen-Namen und Texte sind pro Befehl konfigurierbar (Schul-Thema optional).

Spitzenreiter & Tages-Rekord

Wer im Stream den höchsten Wert in der Top-Stufe würfelt, wird Spitzenreiter – der Bot hängt `{schulleiter_suffix}` an. Der **Tages-Rekord** gilt kalendertags über alle Streams hinweg bis Mitternacht.

Nochmal würfeln?

Bei „Einmal pro Stream“ würfelt der Bot nicht neu, sondern erinnert freundlich an den bestehenden Wert – das zählt als normale Bot-Antwort im Protokoll, kein Fehler.

Daten zurücksetzen

- **6 Alles zurücksetzen** – alle Würfe und Zähler (Tests)
- **7 Nur dieser Stream** – Stream-Würfe löschen, alle dürfen neu würfeln
- **8 Nur Tages-Rekord** – Rekord zurück, Stream-Würfe bleiben

Nach „Nur dieser Stream“ muss `!streamstart` erneut laufen – sonst blockiert „Einmal pro Stream“.

Öffentliches Board: `{wtr_url}` oder Link im Panel unter Community / Killer-Board.

B13. Watchtime

Menü **Bot** → **Watchtime**. Trackt Chat-Anwesenheit pro Zuschauer – **kein** vollständiger Chat-Log.

Deploy-Schutz: Nach Bot-Neustart oder Deploy bleiben laufende Stream-Sessions für ca. `STREAM_BOOT_GRACE_SEC` Sekunden (Standard 120 s) erhalten – Watchtime-Timer und `!streamstart` werden nicht fälschlich beendet. Die Downtime wird der Session gutgeschrieben (bis max. 2 h). Community-Punkte aus Watchtime speichern Restsekunden über Neustarts hinweg.

Tabs im Panel

Tab	Inhalt
Leaderboard	Rangliste, Suche, manuelle Korrektur (+/– Zeit), Einträge löschen
Einstellungen	Tick-Intervall (1s / 1min / 10min), ignorierte Logins (Bots), Vergleichs-Sprüche (Gemini)
Import	StreamElements JWT, CSV (StreamLabs/SE), Vorschau vor Import

Chat-Befehle

- **9** `watchtime` / `watchtime_rank` – eigene Zeit bzw. Platz (Trigger frei wählbar, z. B. `!wt` , `!wtr`)
- **10** Bausteine: `{watchtime}` , `{watchtime_detailed}` , `{rank}` , `{total_users}` , Zeit-Anteile `{hours}` ...
- **11** Öffentliches Board: `/watchtime?channel=LOGIN` · OBS: `/overlay/watchtime?channel=LOGIN&limit=10`

StreamElements-Import

- 1** Bei StreamElements: JWT-Token kopieren (Account → API)
- 2** Im Tab **Import** einfügen → **Token speichern**
- 3** **Vorschau** prüfen → **Alle Watchtime importieren**
- 4** Bestehende Zeiten pro User werden überschrieben – vorher exportieren wenn nötig

Datenexport (Migration & Backup)

Vor dem Import oder beim Bot-Wechsel: Watchtime und andere Bot-Daten unter **Bot** → **Datenexport** sichern — Details in Abschnitt **13c**. Im Tab Import gibt es außerdem den Export des **letzten Streams** als CSV.

Gemini-Vergleichs-Sprüche (optional)

Beim Watchtime-Vergleich zweier User kurze KI-Sprüche – eigener Google-Gemini-API-Key im Panel, Modell wählbar. Ohne Key: statische Fallback-Texte.

Ignorierte Logins

StreamElements, Pokémon Community Game, eigener Bot etc. bekommen keine Watchtime. Der Kanal-Bot wird zusätzlich automatisch ignoriert.

B13b. Community-Punkte

Menü **Bot** → **Community-Punkte**. Internes Punktesystem – **nicht** dasselbe wie Twitch-Kanalkpunkte.

- **5** Modul unter **Module** aktivieren (`community_points`)
- **6** Punkte verdienen: Watchtime (Intervall im Panel, z. B. 1 Punkt alle 5 Min. – **Watchtime-Modul muss aktiv sein**) und/oder Chat-Nachrichten (Cooldown, optional nur live)
- **7** Watchtime-Punkte sammeln Restsekunden zwischen; nach Deploy/Neustart geht der Fortschritt nicht verloren
- **8** Befehle: `points` / `points_rank` – Trigger frei (Standard `!punkte` , `!top`)
- **9** Bausteine: `{points}` , `{currency}` , `{rank}` , `{total}`
- **10** Mods: `!punkte give @user 100` , `take` , `set`
- **11** Panel: Rangliste durchsuchen, Punkte pro User bearbeiten (geben/nehmen/setzen), Sperren/Entsperren
- **12** Öffentliches Board: `/punkte?channel=LOGIN` — Rangliste und Shop-Vorschau; Baustein `{points_url}`

Sperren

Im Panel kann der Streamer Zuschauer für Community-Punkte sperren (1h, 24h, 7 Tage oder permanent). Gesperrte User verdienen keine Punkte und erhalten bei `!punkte` , `!top` , `!shop` und `!kaufen` eine Hinweis-Nachricht. Mod-Anpassungen im Panel bleiben möglich.

Shop (optional)

Im Panel **Shop aktivieren** – dann zusätzlich:

- **13** Artikel anlegen: ID, Titel, Kosten, Aktion (Chat, Sound über OBS aus eigener Bibliothek/Pool, oder nur Abzug)

- **14** Befehle: `points_shop / points_buy` (Standard `!shop` , `!kaufen sound1`)
- **15** Limits: pro Stream (gesamt), pro User (gesamt), pro User pro Stream, Cooldown – Einlösung wird im Punkte-Log gespeichert
- **16** Bei Fehlern (z. B. Sound-URL fehlt) werden Punkte automatisch zurückerstattet
- **17** Shop nur im Chat einlösbar — die öffentliche Seite zeigt Artikel und Preise zur Ansicht
- **18** Export: Salden und Buchungsverlauf unter **Bot → Datenexport** (Abschnitt 13c)

B13bb. Belohnungen

Menü **Bot → Belohnungen**. Eigenes Rewards-Modul für Daily Login und einen Belohnungskatalog.

- **19** Modul unter **Module** aktivieren (`rewards`)
- **20** **Daily Login**: reiner Streak-Zähler — einmal pro Person pro Stream. Ein ausgelassener Stream setzt die Serie zurück (streambasiert, nicht pro Kalendertag). Gespeichert werden aktuelle Streak und längste Streak (Rekord).
- **21** **Check-in-Wege (frei kombinierbar)**: Chat-Befehl (z. B. `!dailylogin`) und/oder Twitch-Kanalpunkte-Einlösung — beide teilen sich dieselbe Streak (1× pro Person pro Stream)
- **22** **Twitch optional**: Zuschauer lösen eine Kanalpunkte-Belohnung ein — der Bot antwortet im Chat und zählt nur die Streak
- **23** **Ohne Community-Punkte**: Der Daily Login ist vom Punktesystem entkoppelt — es gibt keine Punkte-Gutschrift und keine Einlösung über den Community-Shop; der Check-in zählt ausschließlich die Streak
- **24** **Öffentliche Rangliste**: zeigt aktuelle Streak, Rekord (längste Streak), Flammen und Meilenstein-Badges; die eigene Position lässt sich per Suche finden
- **25** **Öffentliches Board**: `/daily?channel=LOGIN` — Streak-Rangliste für Zuschauer; Link im Panel unter Belohnungen

B13c. Bot-Datenexport

Menü **Bot → Datenexport**. Im Gegensatz zu vielen Mitbewerbern kannst du Zuschauer-Daten aus Streamheim **einzelne als CSV** oder **gesamt als ZIP** herunterladen — für Backup, Excel-Auswertung oder Wechsel zu anderen Bots.

Was kann exportiert werden?

Export	Inhalt	Formate
Watchtime (Gesamt)	Alle Zuschauer mit Watchtime, Rang	Streamheim-CSV, StreamElements (<code>username;minutes</code>), einfach (<code>login;seconds</code>)
Watchtime (letzter Stream)	Recap inkl. reiner Chat-Anwesenheit	CSV (auch unter Watchtime → Import)
Community-Punkte	Salden inkl. Lifetime earned/spent	CSV
Punkte-Verlauf	Buchungen aus dem Ledger (bis 50.000 Einträge)	CSV
Chat-Duell	Siege, Niederlagen, Runden gewonnen	CSV
Killer / Würfel	Rollen, Bestwerte, Schulleiter-Zähler	CSV (falls Modul genutzt)
Gesamt-Export	Alle vorhandenen Datensätze + README	ZIP

Technik & Hinweise

- **26** CSV: Semikolon-Trennung, UTF-8 mit BOM (Excel-kompatibel)
- **27** Auch **deaktivierte Module** werden exportiert, solange noch Daten in der Datenbank sind
- **28** Jeder Export wird im **Protokoll** (Audit) vermerkt
- **29** Berechtigung: **Bot-Steuerung** (Ansehen reicht für Download)
- **30** Re-Import nach Streamheim: Watchtime unter **Bot → Watchtime → Import**

Datenschutz: Du lädst Zuschauer-Daten auf dein Gerät. Wenn du sie an andere Dienste weitergibst, informiere deine Community — Details in der Datenschutzerklärung (Abschnitt Bot-Datenexport).

B14. Chat-Duell

Menü **Bot** → **Chat-Duell**. Schere-Stein-Papier-Echse-Spock (SSPS) im Chat – zwei Zuschauer duellieren im Best-of-Modus. Der Bot würfelt die Züge und wertet jede Runde aus.

Ablauf im Chat

0. **31** `!duell @Gegner` — Spieler A fordert B heraus
0. **32** B antwortet mit `!duell` (Annahme) oder `!duell nein` (Ablehnung)
0. **33** Beide senden `!zug` — der Bot wählt pro Person **zufällig** einen Zug (kein Zugname nötig)
0. **34** Wer zuerst die konfigurierte Rundenanzahl gewinnt, siegt im Match und erscheint in der Bestenliste

Befehl	Funktion
<code>!duell @Gegner</code>	Herausforderung senden
<code>!duell</code>	Herausforderung annehmen
<code>!duell nein</code>	Herausforderung ablehnen
<code>!duell abbrechen</code>	Eigene offene Herausforderung zurückziehen
<code>!duell rang</code>	Bestenliste im Chat anzeigen
<code>!zug</code>	Zug wählen — Bot würfelt aus Schere, Stein, Papier, Echse, Spock (Liste im Panel anpassbar)

Zeitlimits

Einstellung	Standard	Bedeutung
Herausforderung läuft ab	120 s	Gegner muss in dieser Zeit mit <code>!duell</code> antworten, sonst verfällt die Einladung
Zug-Timeout	90 s	Nach dem ersten <code>!zug</code> einer Runde muss der Gegner ebenfalls ziehen
Erinnerung nach	45 s	Bot erinnert den fehlenden Spieler noch einmal im Chat
Runden zum Sieg	3	Best-of-3 — im Panel 1-9 einstellbar

Zug-Timeout — faire Auswertung

Wenn ein Spieler nach der Erinnerung immer noch nicht `!zug` sendet, endet das Duell automatisch. Damit niemand absichtlich wartet und trotzdem gewinnt, gilt:

- **35 Pünktlicher führt im Punktestand** → er gewinnt das Match (Sieg in der Bestenliste)
- **36 Unentschieden oder Langsamer führt** → Duell endet *ohne Sieger* (kein Eintrag, kein Gewinn durch Timeout)

Beispiele bei Best-of-3: Stand 2:1, A hat gezogen, B nicht → A gewinnt. Stand 1:2, A wartet, B zieht nicht → Abbruch ohne Sieger.

Chat-Vorlagen: Alle Bot-Nachrichten (Herausforderung, Runde, Erinnerung, Timeout-Sieg, Timeout ohne Sieger, Sieg ...) sind im Panel unter **Einstellungen → Chat-Texte** anpassbar. Bausteine u. a. `{challenger}` , `{winner}` , `{missing}` , `{score_a}` , `{score_b}` , `{seconds_left}` .

Panel-Funktionen

- **37** Laufendes Duell live sehen, im Panel abbrechen (optional Chat-Hinweis)
- **38** Trigger für Herausforderung und Zug frei wählen (z. B. `!ssps` statt `!duell`)
- **39** Zugliste und Schlag-Regeln bearbeiten (Standard: SSPS)
- **40** Chat-Vorlagen pro Ereignis mit Bausteinen
- **41 Duell-Sperren** im Modul – *keine* Twitch-Bans, nur Spielregeln für `!duell`
- **42** Bestenliste + öffentliches Board: `/duell?channel=LOGIN`
- **43** Reset der Rangliste mit optionalem Chat-Ankündigung
- **44** Bestenliste exportieren: **Bot → Datenexport** (Abschnitt 13c)

Wenn **StreamElements** oder ein anderer Bot ebenfalls auf `!duell` reagiert, dort das Duell-Modul deaktivieren oder im Panel einen anderen Trigger wählen.

B14b. Community-Build

Menü **Bot → Community-Build**. Die Community baut im Chat ein gemeinsames Projekt — z. B. Sandburg, Blumenwiese oder Rieseneis. Ziel ist **Zusammenhalt**, kein Wettkampf: Es gibt **keine öffentliche Rangliste** und kein Einzel-Ranking im Chat.

Vorlagen & Start

- 1** Modul **Community-Build** unter **Bot → Module** aktivieren
- 2** Im Panel eine **Vorlage** wählen oder **Eigenes Projekt** anlegen (Name, Emoji, Einheit, Trigger)

3 Mod startet mit `!{steuer} start` — z. B. `!sandburg start`

4 Chat hilft mit `!{mitmach}` — z. B. `!bau`, `!giessen`, `!toppings`

Vorlage	Steuer / Stand	Mitmachen	Einheit
☐ Sandburg	<code>!sandburg</code> · <code>!sandburg start</code>	<code>!bau</code>	Sandhände
☐ Blumenwiese	<code>!wiese</code> · <code>!wiese start</code>	<code>!giessen</code>	Tropfen
☐ Rieseneis	<code>!eis</code> · <code>!eis start</code>	<code>!toppings</code>	Toppings
☐ Eigenes Projekt	frei wählbar	frei wählbar	frei wählbar

Steuer- und Stand-Befehl dürfen gleich sein: `!sandburg` zeigt den Fortschritt, `!sandburg start` startet ein neues Projekt. Alle Trigger sind im Panel unter **Chat-Befehle** pro Vorlage anpassbar.

Befehle im Chat

Befehl	Wer?	Funktion
<code>!{steuer} start</code>	Mod (oder wie konfiguriert)	Neues Projekt starten
<code>!{steuer} pause</code> / <code>stop</code> / <code>reset</code>	Mod	Pausieren, beenden oder zurücksetzen
<code>!{steuer}</code>	Alle	Aktuellen Fortschritt anzeigen (eigener Cooldown)
<code>!{mitmach}</code>	Alle	Beitrag leisten — Erfolg per Zufall, Punkte variieren

Du musst **keine** eigenen Einträge unter **Chat-Befehle** anlegen — das Modul fängt die Trigger selbst ab (wie beim Chat-Duell).

Spielregeln (Panel)

- 5 **Ziel:** Gesamtpunkte bis 100 % (Standard 2.000 Einheiten — anpassbar)
- 6 **Erfolgschance:** Wahrscheinlichkeit, dass ein `!{mitmach}` Punkte bringt (Standard 40 %)
- 7 **Punkte pro Treffer:** Mindest- und Höchstwert (z. B. 1–4 Sandhände)
- 8 **Cooldown:** Abstand zwischen Mitmach-Versuchen pro Person
- 9 **Meilensteine:** Chat-Hinweise bei 5 %, 10 %, ... 100 %
- 10 **Stream offline:** optional pausieren statt weiterlaufen lassen

Panel & Live-Vorschau

- **11** Oben: **Live-Karte** mit Fortschritt, Status und Mitwirkenden-Zähler
- **12** **Texte** für Start, Erfolg, Fehlschlag, Meilenstein, Fertigstellung und Status — mit Bausteinen aus der Gruppe *Community-Build*
- **13** Bei aktivem oder pausiertem Build aktualisiert sich die Vorschau per Socket; sonst alle paar Sekunden per Abruf

Wenn ein anderer Bot auf dieselben Trigger reagiert, dort das Modul deaktivieren oder im Panel andere Trigger wählen.

B14c. Galgenmännchen

Menü **Bot → Galgenmännchen**. Die Community rät gemeinsam ein Wort oder eine Phrase. **Start nur im Panel** (nie per Chat-Befehl – sonst wäre das Wort spoilernd). Leben sind **geteilt**; wer das ganze Wort zuerst trifft, gewinnt.

Ablauf

0. **14** Im Panel **eigenes Wort** oder **Zufallswort** starten – Bot kündigt maskiert an
0. **15** Chat rät: `!rate A` (Buchstabe) oder `!rate Wort` (Volltreffer)
0. **16** Falscher Buchstabe oder falsches Wort → +1 gemeinsamer Fehler
0. **17** Schon geratener Buchstabe → Hinweis, kein Fehler (Antworten drosselbar)
0. **18** Leben leer → Wort wird aufgelöst; Panel kann jederzeit abbrechen

Bestenliste

Siege werden getrennt gezählt: **Buchstaben** = der letzte korrekte Buchstabe füllt die Maske (kein Wort-Volltreffer nötig) und **Wort** = Volltreffer per `!rate Wort`. Rang nach Gesamt; bei Gleichstand zählen mehr Wort-Siege. In Panel und Board erscheint zusätzlich das `{ ' }` **zuletzt gelöste Wort**. Eine Legende auf der öffentlichen Seite erklärt die Spalten.

- **19** Chat: `!rate rang` (oder `bestenliste / top`)
- **20** Öffentliches Board: `/c/<slug>/galgen` (Portal-Form wie Watchtime)
- **21** Panel: Top-10, Reset, Link zur öffentlichen Seite

Blind-Start (Mitspielen)

Unter **Zufallswort** wählt der Bot ein Wort. Standard ist die **Streamheim-Standardliste** (mitgeliefert, im Panel einsehbar) — das ist keine eigene Liste. Optional legst du unter Einstellungen **eigene Wortlisten** an. Die Lösung erscheint **nicht** im Panel — Streamerin und Mods können im Chat mitraten. Kürzlich genutzte Wörter werden optional vermieden.

Chat-Spam & OBS

- **22** Treffer/Falsch-Ankündigungen drosselbar (jeden Tipp / Meilensteine / nur Overlay)
- **23** OBS-Browser-Quelle: `/overlay/hangman?channel=...` — Maske, Leben, geratene Buchstaben

Befehl	Funktion
<code>!rate Buchstabe</code>	Einen Buchstaben raten (Trigger im Panel änderbar)
<code>!rate Wort</code>	Ganzes Wort / Phrase raten – Treffer = Wort-Sieg
<code>!rate rang</code>	Top-5 der Bestenliste im Chat

Panel: Trigger, max. Fehler (3-12), Chat-Texte mit Bausteinen (`{mask}` , `{wrong}` , `{guess_trigger}` , `{word}` , `{author}` ...). Eigene Texte für **richtig**, **Buchstabe falsch** und **Wort falsch** (z. B. „ney falsch“). Live-Stand auch unter **Live jetzt** inkl. Abbruch.

B15. Shoutout, Sound, Begrüßungen & Lurk

Shoutout (`!so Name`)

Mods empfehlen andere Streamer. Der Bot holt Anzeigenname und zuletzt gespieltes Spiel von Twitch (Broadcaster-OAuth nötig). Bausteine: `{streamer}` , `{game_played}` , `{streamer_login}` .

Sound-Overlay & Sound-Bibliothek

Sounds laufen **in OBS** (Browser-Quelle), nicht auf dem PC der Zuschauer. Einmal einrichten, danach per Chat-Befehl oder Community-Punkte-Shop auslösen.

- 0. **24** **OBS:** Browser-Quelle mit URL aus **Bot → Sounds** oder **Glücksrad → OBS Browser-Quellen** (`/overlay/sound?channel=LOGIN`), Hintergrund transparent.
- 0. **25** **Audio wählen:** Unter **Bot → Sounds** hochladen, aus dem **Streamheim-Pool** (von anderen Mandanten freigegeben) wählen oder im Befehl eine externe `https://` -URL eintragen.
- 0. **26** **Chat-Befehl** vom Typ **Sound** anlegen (Menü Chat-Befehle) — Sound zuweisen, Lautstärke 0-1, Chat-Antwort optional (leer = nur Sound).

Eigene Bibliothek (pro Mandant)

- **27** Menü **Bot → Sounds** — Upload vom PC (kein eigener Webservice nötig)
- **28** **Kategorien:** *Sound-Schnipsel* (kurze Effekte, max. 3 MB) und *Musik* (längere Tracks, max. 8 MB) — erleichtert Sortierung im Pool

- **29** Anhören, Kategorie ändern, löschen

Streamheim-Pool (optional, plattformweit)

- **30** Haken „**Für alle Streamheim-Mandanten freigeben**“ beim Upload oder bei jedem eigenen Sound — erscheint dann im Pool für *alle* eingeloggten Mandanten im Panel (nicht öffentlich im Internet)
- **31** Nur freigeben, wenn du die Rechte hast (eigene Aufnahme, Lizenz, freie Nutzung); Freigabe jederzeit widerrufbar
- **32** Pool filterbar nach Kategorie (Schnipsel / Musik) und sortierbar (Name, Streamer, neu)
- **33** Im Befehl-Dialog: Quelle **Eigene · Streamheim-Pool · Externe URL**

Community-Punkte-Shop

Shop-Artikel können als Aktion **Sound (OBS)** einen Eintrag aus eigener Bibliothek oder Pool nutzen — gleiche OBS-Quelle nötig.

Chat-Begrüßungen

Unter **Live-Steuerung**: automatische Nachrichten bei Follow, Sub, Gift, Prime, Erstem Chat im Stream sowie wenn jemand die **Zuschauerserie** im Chat teilt (z. B. „3-Streams-Serie“).

Vorlagen mit Bausteinen – kein manueller !-Befehl nötig.

Ereignis	Bausteine	Hinweis
Erstchat, Follow, Sub ...	{author} , {recipient} , {months} , {gift_count}	{user} = Alias für Autor
Zuschauerserie geteilt	{author} , {streak} , {channel_points}	EventSub <code>channel.chat.notification</code> – Bot braucht <code>user:read:chat</code> , nur wenn der Zuschauer aktiv teilt

Lurk

Klassischer Lurk-Modus unter **Live-Steuerung** (Modul **Lurk**): Zuschauer melden sich ab – der Bot speichert das. Schreibt die Person danach wieder, kommt automatisch „Willkommen zurück“. **Trigger und Chat-Texte sind frei wählbar** unter Bot → Befehle (Standard `!lurk` / `!unlurk` , z. B. auch `!afk` / `!back`). Bei Stream-Ende können alle Lurk-Status geleert werden.

B16. Live jetzt

Menü **Bot → Live jetzt** – zentrale Ops-Fläche **während** des Streams. Panel-Recht: **Live jetzt & Steuerung** (`bot_controls`). Schnell senden / Freitext braucht zusätzlich **Chat als Bot** (`bot_chat`) – Standard nur Streamerin, Leitende Mods nur nach Freigabe in Rechte & Rollen. Auf dem Dashboard gibt es einen kurzen Teaser mit Link hierher.

- **34 Stream** – live/offline, Zuschauer, Titel/Spiel
- **35 Stream-Timer** – Uptime, Start/Stop (wenn Modul aktiv)
- **36 Lurk** – Namen der aktiven Lurker, alle löschen
- **37 Sounds** – Now Playing + Warteschlange (Skip/Leeren), solange die OBS-Browser-Quelle offen ist
- **38 Gewinnspiel** – Status + Link zur Auslosungsseite
- **39 Chat-Duell** – offene Herausforderung oder laufendes Match live (Score/Runde), Abbrechen mit Chat-Hinweis
- **40 Galgenmännchen** – Maske, Fehlerstand, Abbrechen; Start unter Bot → Galgenmännchen
- **41 Schnell senden** – Vorlagen und Freitext (Recht `bot_chat`)

Einstellungen (Texte, Intervalle, Lurk-Vorlagen, Sound-Bibliothek) bleiben unter **Live-Steuerung** bzw. den Modul-Seiten – hier nur Ops.

B16b. Live-Steuerung

Menü **Bot → Live-Steuerung** – Einstellungen und Vorlagen. Panel-Recht: **Live jetzt & Steuerung** (`bot_controls`). Module hier: Stream-Timer, Auto-Streamstart, Chat-Begrüßungen, Lurk. „In Chat senden“ bei Schnell-Nachrichten braucht `bot_chat` . Für den laufenden Stream besser Live jetzt nutzen.

Stream-Aktionen (Panel-Buttons)

- **42 Stream starten / stoppen** – wie `!streamstart` / `!streamstop`
- **43 Countdown** – Dauer eingeben (z. B. 5m), Updates im Chat
- **44 Uptime anzeigen** – sofortige Chat-Antwort

Schnell-Nachrichten

Vordefinierte Texte mit einem Klick in den Chat – z. B. „Pause“, „Technik-Problem“, Social-Link. Mit Bausteinen editierbar.

Automatische Nachrichten

- **45** Intervall (z. B. alle 15 Min.), Text-Vorlage, ein/aus
- **46** Scheduler starten/stoppen im Panel
- **47** Option **nur während Stream** – wartet auf `!streamstart`
- **48** Countdown „nächste in ...“ live im Panel

Auto-Streamstart

Eigene Start-/Stop-Befehle bei Twitch live/offline – Trigger und Antwort-Vorlagen pro Kanal.

Chat-Begrüßungen

Follow, Sub, Gift, Prime, Erstchat, geteilte Zuschauerserie – je Ereignis eigene Vorlage, Modul **Chat-Begrüßungen** muss aktiv sein. Für Zuschauerserien: EventSub aktiv, Bot mit Berechtigung `user:read:chat` (unter Setup neu verbinden, falls nötig).

Lurk

Zuschauer melden sich ab und werden begrüßt, wenn sie wieder schreiben. **Trigger frei wählbar** unter Bot → Befehle (Standard `!lurk` / `!unlurk`). Modul und Willkommen-zurück-Text unter Live-Steuerung.

B17. Stream-Timer, OBS & Overlays

Stream-Timer-Befehle

Befehl	Funktion
<code>!streamstart</code>	Startet Uptime und Stream-Zähler
<code>!streamstop</code>	Beendet Stream-Logik, optional Zusammenfassung
<code>!uptime</code>	Zeigt aktuelle Laufzeit
<code>!countdown 5m</code>	Countdown starten – Updates automatisch im Chat
<code>!countdown stop</code>	Countdown abbrechen

Auto-Streamstart

Start- und Stop-Befehle werden automatisch ausgeführt, wenn Twitch live/offline meldet – eigene Trigger pro Kanal in Live-Steuerung konfigurierbar.

OBS-Szenen-Timer (Menü OBS-Timer)

Countdown-Uhren in OBS – je Timer eigene Browser-Quelle-URL.

Feature	Details
Timer-Aktionen	Zeit +/–, setzen, Start, Pause, Reset
Event-Trigger	Follow, Sub (T1–T3), Gift Sub, Raid, Kanalpunkte
Chat-Befehl	Standard <code>!zeit</code> – konfigurierbar
OBS-WebSocket	Verbindung im Setup für Szenen-Wechsel optional

Overlay-URLs (Setup → OBS Browser-Quellen)

Overlay	URL
Glücksrad	<code>/overlay?channel=LOGIN</code>
Sound	<code>/overlay/sound?channel=LOGIN</code>
Watchtime Top-10	<code>/overlay/watchtime?channel=LOGIN&limit=10</code>
Galgenmännchen	<code>/overlay/hangman?channel=LOGIN</code>
Szenen-Timer	<code>/overlay/timer?channel=LOGIN&id=TIMER_ID</code> (pro Timer)
Watchtime-Board	<code>/watchtime?channel=LOGIN</code>
Würfel-Board	<code>/killer?channel=LOGIN</code>
Duell-Board	<code>/duell?channel=LOGIN</code>
Daily-Login-Board	<code>/daily?channel=LOGIN</code> (Streak-Rangliste)
Streamplan	<code>/c/SLUG/schedule</code> bzw. <code>/schedule?channel=LOGIN</code>
Community-Punkte	<code>/punkte?channel=LOGIN</code> (Rangliste + Shop-Vorschau)

Als Browser-Quelle in OBS – Hintergrund transparent, empfohlene Größe steht unter jeder URL im Panel.

B18. Stream-Statistik (Recap)

Menü **Bot → Stream-Statistik** – Auswertung pro abgeschlossenem Stream.

- **49 Watchtime-Podium** – Top-3 Zuschauer des Streams
- **50 Top-Befehle** – meistgenutzte !-Befehle mit Zähler

- **51 Würfel-Highlights** – Würfe, Spitzenreiter, beste Werte
- **52 Spiel-Segmente** – Kategorie-Wechsel alle 5 Min. (Twitch Helix, ab nächstem Stream)
- **53 Trends** – Vergleich zu vorherigen Streams (Streams/Woche, Ø-Zuschauer ...)

B19. Community & Streamer-Verzeichnis

Im **Dashboard** (Bereich Community & Sichtbarkeit) – wenn Modul Watchtime oder Würfel aktiv.

Community-Boards

Automatische Links für Chat und Socials – Watchtime-, Würfel-, Duell-, Daily-Login- und Community-Punkte-Board je nach aktiven Modulen. URLs kopierbar.

Streamer-Verzeichnis (freiwillig)

- 1** Schalter **Gelistet** aktivieren
- 2** Banner hochladen (max. 2 MB, JPEG/PNG/WebP/GIF)
- 3** Kurzbeschreibung (max. 500 Zeichen), Lieblingsspiele (IGDB-Suche), Social-Links
- 4** Speichern → Profil unter `/c/DEIN_LOGIN` und in `/streamer`

Opt-out jederzeit – kein Listing ohne aktiven Schalter. Aktionen erscheinen im Protokoll (Panel).

Auszeichnungen (vom Plattform-Inhaber)

Der **Plattform-Inhaber** kann Mandanten freiwillige **Labels mit Bild und Text** zuweisen (z. B. „Mandant der ersten Stunde“, „Streamheim-Verdienste“). Du als Streamer legst diese nicht selbst fest — sie werden dir im Owner-Panel zugeordnet.

- **5** Sichtbar nur, wenn dein Kanal im Verzeichnis **opt-in** ist (`/streamer` , `/c/DEIN_LOGIN`)
- **6** Anzeige als hervorgehobene Badges auf der Profilkarte und in der Streamer-Liste; Besucher können nach Auszeichnung filtern
- **7** Enthalten: Bezeichnung, optional Kurzbeschreibung (Tooltip), optional Icon/Bild (vom Inhaber hochgeladen)
- **8** Kein Einfluss auf Bot-Rechte oder Modul-Funktionen — reine Hervorhebung in der Community-Ansicht

Details zur Datenverarbeitung: Datenschutzerklärung (Abschnitt Streamer-Verzeichnis).

B19b. Streamplan

Menü **Bot** → **Streamplan** (`/admin/bot/streamplan`). Hier trägst du ein, wann du streamst — als **Einzeltermin** oder **wöchentliche Serie**. Modul **Streamplan** zuvor unter **Bot** → **Module** aktivieren (plattformweit kann Streamheim das Modul in Wartung setzen).

Termine anlegen & verwalten

- **9** **Wöchentlich** mit **mehreren Wochentagen** in einem Termin (z. B. Mo + Mi + Fr) und optionalem **Wiederhol-Ende** („wiederholen bis ...“)
- **10** **Über Nacht & Mehrtages**: 20:00 → 06:00 oder 24h/48h-Streams werden über alle betroffenen Tage angezeigt
- **11** Spiele aus der **Twitch-/IGDB-Datenbank** (Cover & Infos); Showcase-Spiele als Schnellvorschläge; optional mit einem Klick **ins Showcase-Profil übernehmen**
- **12** **Besonderer Stream hervorheben** (Event, Geburtstag, 24h-Marathon): Symbol + freies Label, leuchtend auf dem Board
- **13** **Partnerstream**: Twitch-Namen eintragen. Ist der Partner bei Streamheim, verlinkt „Partnerstream mit X“ auf sein Profil und er bekommt eine **Einladung**, den Termin in seinen eigenen Plan zu übernehmen; sonst wird auf seinen Twitch-Kanal verlinkt
- **14** Kalender in **Woche** oder **Monat**. Termine per **Drag & Drop** auf einen anderen Tag ziehen (bei Serien fragt ein Menü „nur diese Woche“ oder „ganze Serie“), auf einen Tag klicken → dort anlegen
- **15** Das **---Menü** auf jeder Karte bündelt: Bearbeiten, **Verschieben** (nur diese Ausgabe auf eine andere Zeit), Absagen/Absage zurücknehmen und **ganze Serie löschen**

Öffentliches Board & Kalender-Abo

- **16** Öffentliches Board: `/c/DEIN_SLUG/schedule` (wie Watchtime/Punkte), Fallback `/schedule?channel=DEIN_LOGIN` — Timeline mit „Heute“-Marker
- **17** **Live-Status**: „Jetzt live“ (mit rotem Live-Punkt), wenn du sendest, sonst „beginnt in ...“ für den nächsten Stream
- **18** Klick auf einen Termin öffnet ein **Detail-Panel** inkl. „zu diesem Termin in den Kalender“
- **19** **Kalender abonnieren**: Zuschauer holen sich deinen Plan per Klick in Google/Apple Calendar (`webcal`) oder laden eine `.ics` . Änderungen kommen beim nächsten Abgleich automatisch nach; Absagen verschwinden

Im Chat: `!plan`

Unter **Bot → Befehle** einen Befehl mit Typ „**Nächster Stream (Streamplan)**“ anlegen (Trigger frei, z. B. `plan`). Er postet den nächsten geplanten Stream — online wie offline; steht nichts an, meldet der Bot das automatisch.

- **20** Platzhalter: `{plan}` (fertiger Satz), `{plan_when}`, `{plan_game}`, `{plan_title}`, `{plan_url}`, `{plan_emoji}`, `{plan_label}`, `{plan_partner}`
- **21** Board-Link als Chat-Baustein überall: `{schedule_url}`

Termine ohne Spiel sind möglich. Die Spielesuche braucht die Twitch-App (IGDB) auf dem Server. Ein einzeln verschobener oder abgesagter Serien-Termin ändert nur diese Ausgabe, nicht die Serie.

B20. Multistream & Shared Chat

Menü **Setup → Multistream** – nur bei zwei Mandanten im Panel oder Spiegel-Kanal.

Modus	Verhalten
Einzel-Chat	Bot antwortet nur in deinem Kanal (Standard)
Haupt-Chat	Bei Multistream: Bot antwortet hier, nicht im Zweitkanal
Spiegel-Kanal	Bot antwortet nur im gewählten Haupt-Chat, nicht hier

Bei Twitch Shared Chat (ein Chat, mehrere Streamer) gilt standardmäßig nur dein Kanal. Ohne `chat:edit`-Scope können Antworten kurzzeitig in allen Chats sichtbar sein – Setup-Hinweis beachten, OAuth erneuern.

B21. Discord: Alarme & Community-Server

Menü **Setup → Discord** (wenn der Plattform-Inhaber Discord-Community aktiviert hat).

Bot-Watch per DM (persönlich)

- 1** Bot zum Discord-Server einladen (Link im Panel, falls nötig)
- 2** **Link-Code erzeugen** → in Discord: `/link CODE`
- 3** Abonnenten erhalten persönliche Alarme nur für diesen Kanal (`/botstatus`, `/abos`)

Alarm-Typen (Bot offline, Token abgelaufen ...) konfiguriert der Inhaber unter Plattform → Bot-Alarme.

Community-Server (optional)

Im Panel kannst du der Streamheim-Community auf Discord beitreten:

- **4 Community beitreten** – Einladungslink aus dem Panel
- **5 Live-Ankündigung** – optional: bei `stream.online` Post in `#live`
- **6 Support-Ticket** – privater Kanal oder `/ticket open` betreff:...
- **7** Nach `/link` erhältst du automatisch die Rolle **@Streamer**

Plattform-News (Rundmail) können zusätzlich einmal in `#rundschreiben` erscheinen — parallel zur E-Mail, nicht stattdessen.

B22. Protokoll & Mandanten-Audit

Unter **Protokoll** siehst du dauerhaft gespeicherte Ereignisse für deinen Kanal.

Die Kategorien

Zähler / Filter	Was zählt?
Auslosungen	Glücksrad-Drehungen
Befehle	Erfolgreiche Chat-Befehle inkl. Bot-Antwort
Panel	Panel-Aktionen, DSGVO-Zuordnungen (📄), Timer, Countdown
Gewinnspiele	Start, Auslosung, Nachlosung

Symbole in der Liste

Symbol	Art
🎡	Glücksrad
💬	Chat-Befehl – darunter oft „Bot im Chat: ...“
⚙️	Panel-Aktion
📄	DSGVO (Zuordnung, Antwort, Anfrage)
🎰 / 🎰	Gewinnspiel / Nachlosung

Nur für den Plattform-Inhaber sichtbar

- **8** ⚠️ **Befehl-Fehler** – Befehl existiert nicht, keine Berechtigung, Cooldown, Stream nicht gestartet
- **9** SMTP, Server-Logs, Login-Audit aller Panel-User

Was nicht protokolliert wird

- **10** Normaler Smalltalk („Hi“, „lol“, ...)
- **11** Unbekannte !-Befehle ohne Panel-Eintrag
- **12** Private Nachrichten (Whispers)

Der Bot führt **keine Bans oder Timeouts auf Twitch** aus – Moderation bleibt bei dir und deinen Mods.

B23. Datenschutz im Panel (Self-Service)

Unter **Mein Konto** → **Datenschutz** (nach Login):

Datenexport (JSON - persönlich)

- **13** Profil, Kanal-Mitgliedschaften mit Rollen
- **14 Modul-Status** pro Kanal inkl. letzter Statusänderung
- **15** Login-Verlauf (eigene Anmeldungen)
- **16** E-Mail-Protokoll (an dich gesendete System-Mails)
- **17** Deine DSGVO-Anfragen und Antworten

Bot-Datenexport (CSV/ZIP - Kanal)

Mit Berechtigung **Bot-Steuerung** unter **Bot** → **Datenexport**: Watchtime, Community-Punkte, Chat-Duell, Killer-Statistik — einzeln oder als ZIP. Für Streamer/Mods zum Backup oder Bot-Wechsel, nicht für Endnutzer-Zuschauer. Siehe Abschnitt **13c**.

Weitere Rechte

- **18 Berichtigung:** Name/E-Mail im Profil selbst ändern
- **19 Anfrage stellen:** Auskunft, Löschung, Widerspruch, Einschränkung, Datenübertragbarkeit
- **20 Löschung:** Konto-Löschanfrage mit Passwort-Bestätigung
- **21 Per E-Mail:** dsgvo@streamheim.de

Details und Rechtsgrundlagen: Datenschutzerklärung

B23b. Ideen-Board (Mandant)

Menü **Start** → **Ideen-Board** (`/ideas`) — plattformweit für alle eingeloggten Mandanten mit Dashboard-Zugriff. **Nicht öffentlich** und **kein Ersatz** für DSGVO-Anfragen (dafür **Mein Konto** → **Datenschutz** oder `dsgvo@streamheim.de`).

Abgrenzung: Hier schreiben und abstimmen Mandanten. Die Moderations-Ansicht für den Plattform-Inhaber steht unter D27b · Plattform: Ideen-Board.

Was du dort machen kannst

- **22 Neue Idee** — Titel, Text, Kategorie (Modul-Idee, Bug, Diskussion, Sonstiges)
- **23 Abstimmen** — eine Stimme pro Person und Beitrag (👍)
- **24 Antworten** — Diskussion mit anderen Mandanten; der Plattform-Inhaber antwortet als „Streamheim“
- **25 Sortierung** — zuletzt aktiv, meiste Stimmen oder neueste

Sichtbarkeit & Daten

- **26** Alle Beiträge sind für **alle berechtigten Panel-Nutzer** sichtbar (intern, nicht im Web)
- **27** Gespeichert werden u. a. Anzeigename, zugeordneter Mandant (`@twitch_login`), Titel, Text, Stimmen und Antworten
- **28** Bitte **keine Zuschauer-Daten** und keine sensiblen personenbezogenen Daten Dritter posten
- **29** Bei neuer Idee kann der Inhaber per E-Mail benachrichtigt werden

Plattform-Inhaber (Moderation)

Menü **Plattform** → **Ideen-Board** (`/plattform/ideas`) — zusätzlich:

- **30** Status: Offen → Geplant → In Arbeit → Umgesetzt / Abgelehnt / Geschlossen
- **31 Anpinnen** wichtiger Themen
- **32 Schließen** — keine neuen Antworten (Inhaber kann weiter antworten)
- **33** Statistik: offen, in Arbeit, umgesetzt

Abgrenzung: Feature-Wünsche und Community-Feedback gehören ins Ideen-Board. Auskunft, Löschung, Widerspruch und vergleichbare Rechte nur über Datenschutz (B23) — nicht als normaler Forum-Post.

B24. Was der Bot kann - und was nicht

▢ Das macht der Bot

- **34** Glücksrad-Ergebnis im Chat posten
- **35** Nur auf `!`-Befehle reagieren, die du anlegst
- **36** Timer, Countdown, Uptime, Würfel, Watchtime
- **37** Gewinnspiele, Chat-Duelle, Galgenmännchen, Community-Build, Shoutouts, Begrüßungen, Lurk
- **38** Overlays in OBS triggern (Rad, Sound, Watchtime)
- **39** Per API anbinden (Stream Deck, Streamer.bot) — Setup → API

▢ Das macht der Bot nicht

- **40** Keine privaten Nachrichten lesen
- **41** Keinen normalen Chat archivieren
- **42** Niemanden bannen, timeouten oder modden
- **43** Keine Channel-Einstellungen ändern
- **44** Nur auf deinem eigenen Kanal aktiv
- **45** Kein Zugriff auf Streamer-Passwort oder Stream-Key

Mod-Rechte (`/mod BOT_NAME`) braucht er nur zum Schreiben im Chat – er nutzt keine Mod-Funktionen wie Ban oder Timeout. Den Bot-Namen findest du unter **Setup → Twitch & Discord**.

Öffentliche API

Streamheim ist per **API-Schlüssel** an Stream Deck, Streamer.bot und eigene Webseiten anbindbar — ohne Programmieren, mit fertigen **No-Code-Vorlagen** unter **Setup → API**. Lesen (Status, Watchtime, Punkte, Gewinnspiel), Auslösen (Sound, Rad, Gewinnspiel, Stream-Timer), **Webhooks** (Streamheim meldet sich bei deiner URL) und **Live-WebSocket** für eigene Overlays (Scope `over lay:live`).

Schlüssel: **Setup → API** · Handbuch: panel.streamheim.de/api-handbuch

C25. Öffentliche URLs & Landing

URL	Inhalt
<code>streamheim.de / /</code>	Landing – Produktinfo, Streamer-Verzeichnis-Link
<code>/streamer</code>	Öffentliches Streamer-Verzeichnis (nur Opt-in-Profil)
<code>/c/LOGIN</code>	Einzelprofil eines gelisteten Streamers
<code>/watchtime?channel=...</code>	Watchtime-Bestenliste
<code>/killer?channel=...</code>	Würfel-/Prozent-Board
<code>/c/LOGIN/galgen</code>	Galgenmännchen-Bestenliste (Portal)
<code>/duell?channel=...</code>	Chat-Duell-Rangliste
<code>/daily?channel=...</code>	Daily-Login-Streak-Rangliste
<code>/c/.../schedule</code> bzw. <code>/schedule?channel=...</code>	Streamplan (Woche/Monat)
<code>/punkte?channel=...</code>	Community-Punkte-Rangliste (+ Shop-Vorschau)
<code>/over lay/...</code>	OBS-Browser-Quellen (Rad, Sound, Watchtime, Galgenmännchen, Timer)
<code>/handbuch</code> · <code>/handbuch.pdf</code>	Dieses Handbuch
<code>/api-handbuch</code> · <code>/handbuch/api</code>	API-Handbuch (No-Code, REST v1)
<code>/api/v1/...</code>	Öffentliche REST-API (mit Schlüssel, Setup → API)
<code>/datenschutz</code> · <code>/agb</code> · Impressum	Rechtstexte
<code>/maintenance</code>	Wartungsseite wenn Wartungsmodus aktiv
<code>/admin</code>	Panel-Login

D26. Plattform: Mandanten-Verwaltung

Nur Inhaber – Menü **Plattform** → **Channels** (oder ohne aktiven Kanal nach Login).

Aktion	Beschreibung
Öffnen	In Mandanten-Kontext wechseln (Channel-Switcher)
Mitglieder	Team-Liste, Passwort zurücksetzen (Support)
Datenexport	JSON-Export aller Mandanten-Daten (Art. 15)
Sperren	Mandant deaktivieren – Bot & Panel-Zugang gesperrt, Grund protokolliert
Entsperren	Sperre aufheben
Löschen	Soft-Delete mit Bestätigung – irreversibel nach Frist
Live-Badge	Twitch-Live-Status, Titel, Spiel, Zuschauerzahl

Ereignisse landen im **Mandanten-Audit** (Plattform → Mandanten-Audit) und teils im Mandanten-Protokoll.

D26b. Plattform: Auszeichnungen

Nur Inhaber – Menü **Plattform** → **Auszeichnungen** (`/platform/awards`).

Eigene Kategorien/Labels für Mandanten, die im Streamer-Verzeichnis besonders hervorgehoben werden sollen.

Katalog verwalten

Feld	Bedeutung
Bezeichnung	Anzeigetext im Verzeichnis (z. B. „Mandant der ersten Stunde“)
Beschreibung	Optional — Tooltip auf Profil und in der Liste
Bild	Optionales Icon/Badge (JPEG/PNG/WebP/GIF, max. 512 KB) — wird auf unserem Server gespeichert
Slug	Interne Kennung (URL-freundlich), automatisch aus der Bezeichnung
Sortierung	Reihenfolge bei mehreren Auszeichnungen pro Kanal
Aktiv	Deaktivierte Auszeichnungen werden weder zugewiesen noch öffentlich angezeigt

Mandanten zuweisen

- 1 Mandant in der Dropdown-Liste wählen (Hinweis „im Verzeichnis“, wenn `showcase_opt_in` aktiv)
- 2 Passende Auszeichnungen per Checkbox setzen
- 3 **Zuweisungen speichern** — Zuordnung gilt sofort für die öffentliche Anzeige (sofern Opt-in)

In der Mandanten-Übersicht (**Plattform → Channels**) siehst du zugewiesene Auszeichnungen direkt in der Channel-Spalte.

Öffentliche Wirkung

- 4 `/streamer` — Badges auf den Karten, Filter-Chips nach Kategorie
- 5 `/c/LOGIN` — eigener Abschnitt „Auszeichnungen“ auf dem Profil
- 6 Startseiten-Spotlight — kompakte Badge-Anzeige bei gelisteten Streamern

Zuweisungen werden im **Zugriffs-Audit** (Owner-Protokoll) protokolliert.

D27. Plattform: Bot-Alarme & Discord-Bot

Menü **Plattform → Bot-Alarme**.

Alarm-Kanäle

Kanal	Einstellung
E-Mail	An Streamer, Owner, Extra-Adressen – pro Alarm-Typ
Discord Webhook	Server-Kanal-Benachrichtigungen
SMS (Twilio)	Handy-Nummern für kritische Alarmer
Discord-Bot	Persönliche DMs für verknüpfte Team-Mitglieder (Setup → Discord)

Alarm-Typen (Auszug)

Typ	Wer wird benachrichtigt?
Bot nicht im IRC / Token abgelaufen	Streamer + Owner
Broadcaster-Token abgelaufen	Streamer + Owner
MySQL / PM2 offline	Nur Owner / Ops
Token erneuert, Reconnect	Nur Fehler-Logs (kein Alarm)

Test-Alarm senden zum Prüfen der Konfiguration. Discord-Bot: API-Key, Invite-URL, Enable-Schalter.

Discord-Community (Support-Server)

Menü **Plattform** → **Bot-Alarmer** → **Discord-Community** — Server-ID, Einladungslink, Kanal-IDs, Rollen.

Funktion	Kanal / Rolle
Live-Posts	#live — pro Mandant im Setup aktivierbar
Support-Tickets	Private Kanäle unter SUPPORT · Team: /ticket zuweisen , /ticket schliessen
Rundschreiben	#rundschreiben — optional einmal pro Panel-Rundmail (Embed + @Streamer-Ping; Anhänge nur per E-Mail)
Rollen	@Streamer (automatisch bei /link), @Team (Ticket-Zugriff)

D27b. Plattform: Ideen-Board

Menü **Plattform** → **Ideen-Board** — gleiche Oberfläche wie Mandanten unter `/ideas`, mit Moderations-Werkzeugen und Statistik. Siehe auch B23b.

Funktion	Beschreibung
Übersicht	Alle Mandanten-Beiträge, Filter, Sortierung
Status	Offen, Geplant, In Arbeit, Umgesetzt, Abgelehnt, Geschlossen
Anpinnen	Wichtige Themen oben halten
Schließen	Thread für Mandanten sperren (Support-Antworten weiter möglich)
E-Mail	Benachrichtigung bei neuer Idee (System-SMTP)

Rechtliche Einordnung: internes Panel, nicht öffentlich — Details in der Datenschutzerklärung (Ideen-Board).

D28. Plattform: DSGVO-Postfach

Menü **Plattform** → **DSGVO & Eingänge**.

- 1 E-Mails an `dsgvo@streamheim.de` – IMAP-Import, Auto-Sync ca. alle 3 Min., manuell „Mails abrufen“
- 2 Panel-Anfragen aus „Mein Konto → Datenschutz“ in derselben Liste
- 3 **Zuordnung:** Mandant, optional Panel-User, Anfragetyp (Auskunft, Löschung, Widerspruch ...)
- 4 **Fristen:** 30 Tage – Badges bei bald ablaufend / überfällig
- 5 **Antworten** per SMTP aus dem Panel – signiert mit Bearbeiternamen, Thread-Erhalt
- 6 Status: offen → in Bearbeitung → erledigt; interne Notizen; Auto-Vorschläge bei Zuordnung
- 7 **Löschung ausführen** – Konto anonymisieren (nur bei Lösch-Anfragen)
- 8 **Export** pro User oder Mandant inkl. Modul-Status, Korrespondenz, Login-Audit
- 9 Zuordnungen im **Mandanten-Protokoll** (📄) und Mandanten-Audit

IMAP/SMTP (Plattform)

Postfach-Einstellungen unter DSGVO-Panel: IMAP für Import, SMTP für Antworten – getrennt vom System-SMTP unter „SMTP“.

D29. Plattform: System, SMTP, Logs & Wartung

Menüpunkt	Funktion
System-Health	MySQL, PM2, Bot-IRC, Speicher, Uptime
Wartung	Wartungsmodus für Landing/Panel – öffentliche Hinweis-Seite
Twitch-App	Plattform-OAuth Client-ID/Secret, Redirect-URLs
SMTP	System-Mail (Passwort-Reset, Alarme, Rundmail)
Rundmail	E-Mail an Mandanten; optional parallel ein Post in <code>#rundschriften</code> auf dem Discord-Server
Ideen-Board	Mandanten-Feedback, Diskussion, Moderation
E-Mail-Protokoll	Versand-Log (sent/failed), Typ, Empfänger
Logins	Panel-Anmeldungen aller User (Zeit, IP, Erfolg)
Zugriffs-Audit	Owner-Aktionen: Channel-Wechsel, Sperren, Export, Auszeichnungen ...
Mandanten-Audit	Sperren, Löschen, DSGVO pro Kanal
Fehler-Logs	Server-Debug-Log, filterbar, für Support

D30. Plattform: !dev & Sonstiges

Chat !dev (nur Betreiber)

Menü **Plattform** → **Chat !dev** – erlaubte Twitch-Nicks (ein Name pro Zeile). Nur diese dürfen `!dev` im Chat nutzen – erscheint nicht in der Mandanten-Befehlsliste. Ohne Eintrag: niemand (auch keine Mods).

Owner ohne Kanal

Owner landet ohne gewählten Kanal direkt auf der Plattform-Übersicht. Mit Channel-Switcher in einen Mandanten wechseln für Stream-Setup.

E31. Daten, Sorgfalt & Hilfe

Was gespeichert wird

- **10 Panel-Benutzer:** Login, Anzeigename, gehashtes Passwort, Rolle; E-Mail freiwillig
- **11 Glücksrad / Befehle / Panel:** siehe Protokoll
- **12 Würfel & Watchtime:** Werte für Listen, Rang und Erinnerungen
- **13 Daily Login:** aktuelle Streak, längste Streak (Rekord), letzter Check-in (Datum + Stream) und Check-in-Verlauf pro Login; zusätzlich pro Kanal die Stream-Startzeitpunkte (ohne Personenbezug) für die stream-basierte Serie — optional öffentliche Streak-Rangliste
- **14 Community-Build:** Fortschritt, Mitwirkungszähler pro Login (Versuche/Erfolge) — nur bei aktivem Modul, keine öffentliche Rangliste
- **15 DSGVO:** Anfragen, Zuordnungen, Bearbeitungsstatus (Plattform)
- **16 Ideen-Board:** Beiträge, Antworten, Abstimmungen (Panel-intern, plattformweit unter Mandanten)

Was nicht gespeichert wird

- **17** Normaler Twitch-Chat-Smalltalk
- **18** Viewer-IPs oder Gerätedaten der Zuschauer
- **19** Twitch-Passwörter oder Stream-Keys

Gute Praxis

- **20** Passwörter nicht weitergeben; bei Verdacht unter „Mein Konto“ ändern
- **21** Team-Zugänge deaktivieren, wenn jemand geht
- **22** VIPs nur „Ansehen“, Mods nur was sie brauchen
- **23** Server in Deutschland (Hetzner) – Panel-Daten auf eurem Server

Wer hilft wann?

Situation	Wer?
Befehle, Module, Mods, Rechte, Würfel zurücksetzen	Du / dein Team
Bot antwortet nicht, Setup-Häkchen rot	Setup & FAQ, dann Plattform-Support
Panel-Fehler, E-Mail, DSGVO-Anfragen	Plattform-Inhaber
Feature-Wünsche, Ideen-Board, Produkt-Feedback	Ideen-Board; Moderation durch Inhaber
Neue Features, Server, Updates	Plattform-Inhaber

- **24 Handbuch:** panel.streamheim.de/handbuch
- **25 Support:** kontakt@streamheim.de
- **26 Datenschutz:** dsgvo@streamheim.de
- **27 FAQ im Panel:** Dashboard → „Was kann der Bot?“

Anhang A - Alle Bausteine (Vollständig)

Im Panel unter **Bausteine** als Chips einfügbar. `_next` = mitzählen, ohne `_next` = nur anzeigen.

Personen & Shoutout

Baustein	Bedeutung
<code>{author}</code>	Auslöser (Display-Name); Aliase: <code>{name}</code> , <code>{user}</code>
<code>{target_user}</code>	Ziel bei !befehl @User
<code>{login}</code> <code>{author_login}</code>	Twitch-Login klein
<code>{streamer}</code> <code>{streamer_login}</code> <code>{streamer_url}</code>	Shoutout-Ziel
<code>{game_played}</code> <code>{game}</code> <code>{streamer_plus}</code>	Spiel / Kombi

Zähler

Baustein	Bedeutung
<code>{count}</code> / <code>{count_next}</code>	Gesamt-Nutzungen
<code>{author_count}</code> / <code>{author_count_next}</code>	Pro Autor
<code>{pair_count}</code> / <code>{pair_count_next}</code>	Autor + Ziel
<code>{others_count}</code> / <code>{others_count_next}</code>	Alle außer Autor
<code>{target_count}</code> / <code>{target_count_next}</code>	Pro Ziel

Würfel & Statistik

Baustein	Bedeutung
{random_percent} {random_tier} {roll_icon}	Wurf & Stufe
{schulleiter_suffix} {leader_title} {scope}	Spitzenreiter
{daily_leader_name} {daily_leader_percent}	Tages-Rekord
{board} {klassen_stat} {stats_summary}	Listen & Statistik
{rivalry_line} {rivalry_result}	Stufen-Vergleich
{participant_count} {pct_tier_1} ... {count_tier_3}	Teilnehmer & Stufen-%
{random_number} {random_letter}	Zufall 1-100 / A-Z

Stream, Watchtime, Gewinnspiel, Rad, Links

Baustein	Bedeutung
{uptime} {timer} {countdown}	Stream-Zeiten
{watchtime} ... {seconds} {watchtime_seconds}	Watchtime-Anteile
{rank} {total_users}	Leaderboard
{join} {duration} {winners} {reason}	Gewinnspiel
{prize} {author}	Glücksrad ({user} = Alias)
{url} {wt_url} {wtr_url} {wto_url} {points_url}	URLs (Board/Overlay)
{args} {channel} {limit_when}	Argumente, Kanal, Limit-Text

Community-Build

Baustein	Bedeutung
{project} {emoji} {unit}	Projektname, Emoji, Einheit (Sandhände, Tropfen ...)
{progress} {progress_label} {progress_points}	Fortschritt in % und Einheiten
{goal} {remaining}	Zielpunkte, fehlende Einheiten bis 100 %
{contributors} {points} {author}	Anzahl Mitwirkender, letzter Beitrag, Name
{start_trigger} {contrib_trigger} {status_trigger}	Trigger ohne ! aus den Panel-Einstellungen
{status_label}	aktiv, pausiert, fertig oder inaktiv

Anhang B - Standard-Rechte-Matrix

Voreinstellung - unter Team → Rechte & Rollen anpassbar. ✓ = Bearbeiten, 👁 = Ansehen, - = kein Zugriff.

Bereich	Streamer	Leit. Mod	Mod	VIP
Dashboard, FAQ	✓	✓	✓	–
Glücksrad / Rad / Preise	✓	–	–	–
Chat-Befehle, Bausteine	✓	✓	✓	–
Module, Live jetzt & Steuerung (inkl. Begrüßungen & Lurk), Gewinnspiel, Watchtime, Datenexport, OBS-Timer	✓	✓	–	–
Chat als Bot (Schnell senden)	✓	–	–	–
Chat-Duell	✓	✓	–	–
Galgenmännchen	✓	✓	–	–
Community-Build	✓	✓	–	–
Streamplan	✓	✓	–	–
Protokoll	✓	✓	✓	–
Setup, Multistream, Discord	✓	✓	–	–
Benutzer, Rechte	✓	–	–	–
Mein Konto	✓	✓	✓	✓
System / Plattform	–	–	–	–

Inhaber hat überall Zugriff inkl. Plattform – außer im Mandanten-Kontext ohne explizite Rechte-Matrix-Zeile.