

Media Library

Sound + Bilder + Videos zentral verwalten.

- [Media Library Grundlagen](#)
- [Upload + Tagging](#)
- [Media in Action verwenden](#)

Media Library Grundlagen

Media Library Grundlagen

Die Media Library ist deine zentrale Asset-Verwaltung in Chatlix. Alle Bilder, Sounds und kurzen Videos, die du in Overlays, Automations, Alerts oder Pro-Editor-Widgets nutzt, liegen hier. Pro Account, mit Tags, Kategorien und Such-Funktion.

Das Dashboard erreichst du unter `/dashboard/media`. Der React-Component-Name ist `MediaLibrary`. Backend-seitig steckt das in `Chatlix\Service\MediaService` und unter `Chatlix\Service\Media*` (11 Service-Dateien für Storage, CDN, Transcoding, Asset-Management).

Unterstützte Formate

- **Bilder:** PNG, JPG/JPEG, WebP, GIF (animierte GIFs werden als Animation gespeichert).
- **Audio:** MP3, OGG, WAV.
- **Video:** MP4, WebM. Für längere Aufzeichnungen (Stream-Mitschnitte) ist Clip-Studio das richtige Tool, nicht die Media Library.

Max. Dateigröße standardmäßig 50 MB pro Asset. Bilder werden serverseitig auf Web-tauglichen Formaten gehalten (PNG/JPEG/WebP, GIF bleibt GIF); Audio wird ggf. transcodiert, damit Browser-Wiedergabe konsistent läuft.

Kategorien und Tags

Jedes Asset hat zwei Ordnungs-Dimensionen:

- **Kategorie** (eine pro Asset): `Sound`, `Sticker`, `Alert`, `Background`. Hilft beim schnellen Filtern, wenn du z.B. nur Alert-Sounds suchst.
- **Tags** (mehrere pro Asset, freitext): `air-horn`, `funny`, `intro`, `outro`, `coffee`. Tags sind dein eigenes Suchsystem.

Die Suche oben in der Library greift auf Dateiname, Kategorie und Tags zu.

Plan-Limits

Die Library hat ein **Quota pro Account**. Beispielwerte (in der Web-App genauer angezeigt):

- Free: ~500 MB Gesamtspeicher.
- Pro: deutlich mehr, je nach Plan.
- Bei Erreichen: Upload schlägt fehl mit `MEDIA_QUOTA_EXCEEDED`.

Im Dashboard siehst du oben rechts den belegten/verfügbaren Speicher als Balken. Wenn du nah am Limit bist, lösche zuerst alte/ungenutzte Assets oder upgrade den Plan.

API-Routen

Zwei Route-Familien:

- `/v1/media/` für die Verwaltung:
 - `POST /v1/media/upload` — neues Asset hochladen.
 - `DELETE /v1/media/<id>` — Asset löschen.
 - `GET /v1/media/<id>/proxy` — Asset über Chatlix-CDN streamen (mit Auth, falls nötig).
- `/v1/media-engine/` für Asset-Operationen:
 - `POST /v1/media-engine/play` — vom Server aus ein Sound-Asset in ein Overlay routen.

- `POST /v1/media-engine/probe` — Metadaten extrahieren (Dauer, Codec, Auflösung).
- `POST /v1/media-engine/reprocess` — Transcoding erneut anstoßen, z.B. nach Codec-Update.

CDN und Caching

Assets werden über ein CDN ausgeliefert. Beim Upload bekommst du eine Public-URL (signed, falls die Library privat ist). Im Overlay/Editor läuft das transparent.

Wenn du ein Asset austauscht (z.B. Logo neu hochladen mit gleichem Namen), bekommt es eine neue ID — du musst die alte Referenz manuell ersetzen. Direktes „Überschreiben“ gibt es nicht; das verhindert, dass Live-Streams plötzlich kaputtes Audio bekommen, weil im Hintergrund eine Datei geändert wurde.

Sichtbarkeit

Assets sind standardmäßig **privat**, also nur in deinen Overlays/Automations zugreifbar (über Signaturen). Public-Sharing einzelner Assets ist möglich, aber selten nötig.

Wo verwendet?

Assets werden referenziert in:

- Automations als Action „Sound abspielen“ oder „Bild einblenden“.
- Pro Editor (Overlay-Builder) als Image-, Audio- oder Video-Widget.
- StoryQuest-Nodes als Hintergrund/Sound.
- Alerts als Alert-Sound oder Alert-Image.
- StreamDeck-Buttons als „Sound abspielen“-Aktion.

Wenn du ein Asset löschen willst, das noch irgendwo referenziert wird, zeigt die Library eine Warnliste mit allen Verwendungs-Stellen. Das hilft, kaputte Verweise zu vermeiden.

Upload + Tagging

Upload + Tagging

Dieses Kapitel zeigt, wie du Assets sauber in die Library bekommst und so taggst, dass du sie später wiederfindest.

Upload

Im Dashboard auf `/dashboard/media` klick oben `« Upload »`. Du hast drei Wege:

1. **Drag & Drop**: Dateien aus dem Datei-Explorer auf den Upload-Bereich ziehen. Mehrere gleichzeitig sind ok.
2. **Datei-Auswahl**: Klassischer Datei-Dialog.
3. **URL-Import**: Eine externe URL einfügen; Chatlix lädt das Asset serverseitig und legt es in der Library ab. Nützlich für lizenz-freie Sound-Quellen oder dein eigenes CDN.

Während des Uploads siehst du pro Datei eine Fortschrittsleiste. Nach Abschluss läuft serverseitig die **Probe** (Format-Erkennung, Dauer, Auflösung) und ggf. Transcoding. Bei großen Dateien kann es ein paar Sekunden dauern, bis das Asset spielbar markiert ist.

API-Pendant: `POST /v1/media/upload` als `multipart/form-data` mit `file=<binary>` und optional `name`, `category`, `tags[]`.

Naming

Dateinamen werden bei Upload als Asset-Name übernommen, aber du kannst sie nachträglich umbenennen. Empfehlungen:

- Beschreibend, nicht generisch: `airhorn-loud.mp3` statt `sound1.mp3`.
- Konvention pro Kategorie: z.B. Alert-Sounds mit Präfix `alert-` (`alert-follow.mp3`, `alert-sub.mp3`).
- Keine Sonderzeichen außer `-` und `_`.

Das Erleichtert spätere Sortierung und Suche.

Kategorien setzen

Nach dem Upload öffnest du das Asset (Klick im Grid) und wählst rechts die Kategorie aus dem Dropdown:

- **Sound** — alle Audio-Assets, die in Stream-Situationen abgespielt werden (Alert-Sounds, Soundboard-Effekte).
- **Sticker** — kleine PNGs/GIFs, die als Reaktions-Element eingeblendet werden.
- **Alert** — Alert-Specials (Video-Animations oder Komplett-Alerts).
- **Background** — Hintergründe für StoryQuest-Nodes oder Overlay-Szenen.

Kategorien sind grobe Ordnungs-Hilfen. Wenn keine richtig passt, lass sie leer und nutze Tags.

Tags vergeben

Tags sind das eigentliche Such-Werkzeug. Praxis-Tipps:

- **Konsistent:** Halte eine kleine Tag-Liste, statt für jedes Asset neue Tags zu erfinden.
Beispiele: `intro`, `outro`, `funny`, `dramatic`, `airhorn`, `nope`, `victory`, `defeat`.
- **Mehrdimensional:** Ein Asset kann mehrere Tags haben — Stimmung (`funny`), Verwendung (`alert-follow`), Quelle (`from-stream`).
- **Tools:** Wenn du Assets stapelweise hochlädst, ist im Multi-Select-Modus eine Bulk-Action `« Tags zuweisen »` verfügbar.

Im Such-Feld der Library kannst du nach Tag filtern (z.B. `tag:funny`), nach Kategorie (`cat:Sound`) oder nach Freitext im Dateinamen.

Metadaten prüfen

Im Detail-Panel siehst du nach dem Probe-Run:

- Format, Codec.
- Dauer (bei Audio/Video).
- Auflösung (bei Bild/Video).
- Dateigröße.
- Erstellt-am, Zuletzt-verwendet-am.

Wenn die Dauer 0 ist oder Codec leer, ist die Probe gescheitert. Klick `« Erneut analysieren »` (`POST /v1/media-engine/reprocess`). Bei wiederholtem Fehlschlag: Datei neu encoden (z.B. via ffmpeg) und re-uploaden.

Upload-Fehler

Häufige Codes:

- `MEDIA_QUOTA_EXCEEDED` — Quota voll. Lösche alte Assets oder upgrade.
- `MEDIA_FORMAT_UNSUPPORTED` — Format nicht in Liste (z.B. AVI, MOV). Konvertier zu MP4/WebM.
- `MEDIA_FILE_T00_LARGE` — Über 50 MB. Komprimier oder kürze. Für längere Videos: Clip-Studio nutzen, nicht Media Library.
- `MEDIA_VIRUS_DETECTED` — selten, aber automatisches Scanning hat etwas auffälliges gefunden. Falls du sicher bist, dass die Datei sauber ist, Support melden.

Asset entfernen

Im Detail-Panel . Du bekommst eine Liste aller Verwendungen (Automations, StoryQuests, Overlays). Wenn das Asset noch aktiv genutzt wird, blockiert die Löschung mit der Liste. Erst Referenzen ersetzen, dann erneut löschen.

Wenn das Asset gelöscht ist, ist es weg — keine Wiederherstellung. Bei wichtigen Master-Assets lohnt sich ein lokales Backup deinerseits.

Media in Action verwenden

Media in Action verwenden

Assets aus der Media Library werden an mehreren Stellen referenziert. Hier siehst du die drei wichtigsten: Automations, Pro Editor und StoryQuest.

Im Automation-Action „Sound abspielen“

In `/dashboard/automations` öffnest du eine Automation und klickst `« Action hinzufügen »`. Wähl `Sound abspielen`. Im Config-Panel:

- **Asset:** Asset-Picker, der die Media Library öffnet. Such oder filter nach Kategorie `Sound` und Tag (`alert`, `funny`, ...).
- **Lautstärke:** 0–100% (Standard 80%).
- **Ziel-Overlay:** Welches OBS-Overlay den Sound abspielt. Standard: alle aktiven Overlays.
- **Cooldown:** Sekunden, in denen die Action für denselben User nicht erneut feuert (verhindert Soundboard-Spam).

Speichern, fertig. Im nächsten Trigger (Reward eingelöst, Follower etc.) wird der Sound serverseitig über `POST /v1/media-engine/play` ans Overlay geroutet.

Dasselbe Pattern gilt für `Bild einblenden` (Asset aus Kategorie `Sticker` oder `Alert`) und für Video-Inserts.

Im Pro Editor (Overlay-Builder)

Der Pro Editor in `/dashboard/overlays/<id>` lässt dich Overlay-Layouts pixelgenau bauen. Drei Widget-Typen greifen direkt auf die Library:

- **Image-Widget:** Statisches Bild (Logo, Frame, Wasserzeichen). Asset-Picker öffnet die Library.
- **Audio-Widget:** Sound-Loop oder einmaliger Sound (z.B. Hintergrund-Loop für Pause-Screen).
- **Video-Widget:** Kurzes Video, lokal oder als Loop.

Im Widget-Panel rechts ist der Asset-Picker das erste Feld. Tausch jederzeit möglich; das Overlay re-fetcht den Asset über CDN, kein Reload nötig.

In StoryQuest-Nodes

Im `StoryQuestBuilder` hat jeder Node die optionalen Felder:

- **Hintergrund-Bild:** Asset-Picker auf Bilder. Wird im Overlay als Background-Layer für den Node gerendert.
- **Sound:** Asset-Picker auf Audio. Wird beim Erscheinen des Node abgespielt.

Damit kannst du Atmosphäre pro Szene setzen — z.B. düsterer Wald-Sound für den Wald-Node, knisterndes Feuer für die Lager-Szene.

In Alerts

Unter `/dashboard/alerts` konfigurierst du pro Alert-Typ (Follow, Sub, Cheer, Raid, ...):

- **Alert-Sound:** Asset aus `Sound` oder `Alert`.
- **Alert-Image/Video:** Asset aus `Sticker` oder `Alert`.

- **Alert-Animation:** Falls Asset ein Video oder GIF ist, läuft das als Animation; sonst statisch eingeblendet mit Dauer-Slider.

Die Alert-Engine kombiniert das in eine kurze Sequenz pro Event.

Im StreamDeck

Für Stream-Deck-Buttons mit Aktion `Sound abspielen` (siehe `Verfügbare Aktionen`) wählst du beim Setup ebenfalls direkt aus der Media Library. Der Button triggert dann beim Druck den Sound im Overlay.

API-Pfade

Wenn du ein Asset in einer eigenen Integration nutzen willst (z.B. externes Tool):

- `GET /v1/media/<id>` liefert Metadaten und CDN-URL.
- `GET /v1/media/<id>/proxy` streamt das Asset (Auth nötig).
- `POST /v1/media-engine/play` triggert ein Asset in einem deiner aktiven Overlays:

```
{
  "asset_id": 123,
  "overlay": "main",
  "volume": 0.6,
  "cooldown_seconds": 5
}
```

Verwendungs-Tracking

Die Library zeigt im Detail-Panel eines Assets eine **Verwendungs-Liste**: in welchen Automations/Overlays/Quests/Alerts das Asset referenziert wird. Praktisch, bevor du etwas löschst oder austauschst.

Tipps

- **Loudness-Konsistenz**: Normalisier deine Sound-Assets vor Upload (Ziel -16 LUFS für Spiele, -14 LUFS für Musik). Sonst spielst du teils-leise, teils-laut.
- **Bild-Größen**: Für Overlay-Widgets keine 4K-Bilder hochladen, wenn das Widget 300×300 px groß ist. 2× Display-Auflösung ist genug.
- **Naming-Convention**: Wenn du in Automations wählst, hilft ein klarer Asset-Name.

`alert-follow-default.mp3` ist schneller wiedergefunden als `Recording_2026_05_03_141.mp3`.