

YouTube Data API v3 Integration

La Scribe du Nil — Landingpage für einen YouTube-Kanal

Ausgangslage

Statische HTML/CSS/JS-Website (kein Framework, kein Build-Tool) für einen YouTube-Kanal mit Einschlafgeschichten. Abonnentenzahl und Videoliste waren fest im HTML eingetragen und mussten bei jedem neuen Video manuell aktualisiert werden.

Umsetzung

- Google Cloud Setup: eigenes GCP-Projekt angelegt, YouTube Data API v3 aktiviert, API-Key erstellt und über HTTP-Referrer-Restriction auf die eigene Domain beschränkt (der Key ist zwar öffentlich im Client-Code sichtbar, kann aber nur von der eigenen Domain aus genutzt werden — Standardvorgehen bei client-seitigen YouTube-API-Keys).
- Live-Abonnentenzahl: `channels.list?part=statistics` wird beim Laden der Seite abgefragt und schreibt die aktuelle Zahl in ein `<span id="stat-subs">`. Fallback: Bei Fehler oder falls der Kanal die Abonnentenzahl versteckt hat, bleibt der fest hinterlegte Text im HTML einfach stehen (kein Absturz, kein leerer Zustand).
- Dynamisches Video-Grid: `playlistItems.list` (Uploads-Playlist des Kanals) und `videos.list?part=statistics,contentDetails` holen die neuesten Videos inklusive Views, Likes und Dauer. Die Karten werden clientseitig gerendert und ersetzen das ursprünglich statische Grid.
- Shorts-Filter: Videos unter 60 Sekunden werden anhand des ISO-8601-Dauerformats (z. B. `PT45S` vs. `PT1M5S`) herausgefiltert, damit nur „echte“ Langform-Videos erscheinen.
- Hervorhebung: Das neueste Video erhält eine visuelle Sonderbehandlung (goldener Rahmen, Badge, breiter im Grid auf Desktop).
- Fallback-Strategie: Schlägt ein API-Aufruf fehl, bleibt das ursprüngliche statische HTML unverändert sichtbar — die Seite bricht nie, sie fällt nur auf den zuletzt manuell gepflegten Stand zurück.

Bewirkter Effekt

- Kein manuelles Pflegen des HTML mehr bei neuen Video-Uploads — die Seite zieht sich Inhalte automatisch nach.
 - Zusätzlicher sozialer Beweis durch sichtbare Views und Likes direkt auf der Landingpage.
 - Robuste Fehlerbehandlung: Ein API-Ausfall führt nie zu leeren oder kaputten UI-Zuständen.
-

YouTube Data API v3 Integration

La Scribe du Nil — landing page for a YouTube channel

Starting point

A static HTML/CSS/JS website (no framework, no build tool) for a YouTube channel producing bedtime stories. Subscriber count and video list were hard-coded into the HTML and had to be updated manually with every new video.

Implementation

- Google Cloud setup: created a dedicated GCP project, enabled the YouTube Data API v3, generated an API key and restricted it to the site's own domain via an HTTP referrer restriction (the key is visible in client-side code, but can only be used from the authorized domain — the standard approach for client-side YouTube API keys).
- Live subscriber count: `channels.list?part=statistics` is called on page load and writes the current count into a `<span id="stat-subs">`. Fallback: if the call fails, or the channel has hidden its subscriber count, the hard-coded text already in the HTML simply stays in place (no crash, no empty state).
- Dynamic video grid: `playlistItems.list` (the channel's uploads playlist) combined with `videos.list?part=statistics,contentDetails` fetches the latest videos including views, likes and duration. Cards are rendered client-side and replace the original static grid.
- Shorts filter: videos under 60 seconds are filtered out based on the ISO 8601 duration format (e.g. `PT45S` vs. `PT1M5S`), so only genuine long-form videos are shown.
- Highlighting: the newest video receives special visual treatment (gold border, badge, wider grid placement on desktop).
- Fallback strategy: if an API call fails, the original static HTML remains visible unchanged — the page never breaks, it simply falls back to the last manually maintained state.

Effect

- No more manual HTML maintenance on every new video upload — the page pulls in content automatically.
 - Additional social proof through visible view and like counts directly on the landing page.
 - Robust error handling: an API outage never results in an empty or broken UI state.
-

Intégration de la YouTube Data API v3

La Scribe du Nil — landing page pour une chaîne YouTube

Point de départ

Un site statique HTML/CSS/JS (sans framework, sans outil de build) pour une chaîne YouTube d'histoires pour s'endormir. Le nombre d'abonnés et la liste des vidéos étaient codés en dur dans le HTML et devaient être mis à jour manuellement à chaque nouvelle vidéo.

Mise en œuvre

- Configuration Google Cloud : création d'un projet GCP dédié, activation de l'API YouTube Data v3, génération d'une clé API restreinte au domaine du site via une restriction HTTP referrer (la clé est certes visible dans le code client, mais ne peut être utilisée que depuis le domaine autorisé — pratique standard pour les clés API YouTube côté client).
- Nombre d'abonnés en direct : `channels.list?part=statistics` est appelé au chargement de la page et inscrit le chiffre actuel dans un `<span id="stat-subs">`. Solution de repli : en cas d'erreur, ou si la chaîne a masqué son nombre d'abonnés, le texte codé en dur dans le HTML reste simplement affiché (pas de plantage, pas d'état vide).
- Grille de vidéos dynamique : `playlistItems.list` (playlist des mises en ligne de la chaîne) combiné à `videos.list?part=statistics,contentDetails` récupère les dernières vidéos avec leurs vues, likes et durée. Les cartes sont générées côté client et remplacent la grille statique d'origine.
- Filtre Shorts : les vidéos de moins de 60 secondes sont exclues à partir du format de durée ISO 8601 (par ex. `PT45S` contre `PT1M55S`), afin de n'afficher que les véritables vidéos longues.
- Mise en avant : la vidéo la plus récente bénéficie d'un traitement visuel particulier (bordure dorée, badge, plus large dans la grille sur ordinateur).
- Stratégie de repli : si un appel API échoue, le HTML statique d'origine reste affiché sans changement — la page ne plante jamais, elle revient simplement au dernier état maintenu manuellement.

Effet obtenu

- Plus besoin de maintenir le HTML manuellement à chaque nouvelle vidéo — la page récupère automatiquement le contenu.
 - Preuve sociale supplémentaire grâce aux vues et likes visibles directement sur la landing page.
 - Gestion robuste des erreurs : une panne de l'API ne provoque jamais d'état d'interface vide ou cassé.
-