YouTube Data API – Übersicht

Einführung

Dieses Dokument richtet sich an Entwickler, die Anwendungen für die Interaktion mit YouTube erstellen möchten. Darin werden die grundlegenden Konzepte von YouTube und der API selbst erläutert. Außerdem erhalten Sie einen Überblick über die verschiedenen Funktionen, die von der API unterstützt werden.

Vorbereitung

1. Sie benötigen ein Google-Konto (https://www.google.com/accounts/NewAccount?hl=de), um auf die Google API Console zuzugreifen, einen API-Schlüssel anzufordern und Ihre Anwendung zu registrieren.
2. Erstellen Sie ein Projekt in der Google Developers Console (https://console.developers.google.com?hl=de) und fordern Sie Anmeldedaten für die Autorisierung an (https://developers.google.com/youtube/registering_an_application?hl=de), damit Ihre Anwendung API-Anfragen senden kann.
3. Vergewissere dich nach der Erstellung deines Projekts, dass die YouTube Data API einer der Dienste ist, für den deine Anwendung registriert ist:
    a. Gehen Sie zur API Console (https://console.cloud.google.com/?hl=de) und wählen Sie das Projekt aus, das Sie gerade registriert haben.
    b. Rufen Sie die Seite Aktivierte APIs (https://console.cloud.google.com/apis/enabled?hl=de) auf. Prüfe in der Liste der APIs, ob der Status der **YouTube Data API Version 3 EIN** lautet.
4. Falls deine Anwendung eine API-Methode verwendet, die eine Benutzerautorisierung erfordert, solltest du dir das Handbuch zur Authentifizierung (https://developers.google.com/youtube/v3/guides/authentication?hl=de) durchlesen, um zu erfahren, wie eine OAuth 2.0-Autorisierung implementiert wird.
5. Wähle eine Client-Bibliothek (https://developers.google.com/youtube/v3/libraries?hl=de) aus, um deine API-Implementierung zu vereinfachen.

6. Machen Sie sich mit den Kernkonzepten des JSON-Datenformats (JavaScript Object Notation) vertraut. JSON ist ein gängiges, sprachunabhängiges Datenformat, das eine einfache Textdarstellung beliebiger Datenstrukturen bietet. Weitere Informationen dazu finden Sie unter json.org (http://json.org).

Ressourcen und Ressourcentypen

Eine Ressource ist ein individuelles Datenobjekt mit einer eindeutigen ID. In der folgenden Tabelle werden die verschiedenen Ressourcentypen beschrieben, mit denen Sie über die API interagieren können.

Ressourcen	
`activity`	Enthält Informationen zu einer Aktion, die ein bestimmter Nutzer auf der YouTube-Website ausgeführt hat. Zu den Nutzeraktionen, die in Aktivitätsfeeds gemeldet werden, zählen unter anderem das Bewerten eines Videos, das Teilen eines Videos, das Markieren eines Videos als Favorit und das Posten eines Kanalbulletins.
`channel`	Enthält Informationen zu einem einzelnen YouTube-Kanal.
`channelBanner`	Gibt die URL an, die zum Festlegen eines neu hochgeladenen Bilds als Bannerbild für einen Kanal verwendet werden soll.
`channelSection`	Enthält Informationen zu einer Reihe von Videos, die auf einem Kanal empfohlen werden. In einem Abschnitt können beispielsweise die neuesten Uploads eines Kanals, die beliebtesten Uploads oder Videos aus einer oder mehreren Playlists enthalten sein.
`guideCategory`	Kennzeichnet eine Kategorie, die YouTube anhand der Inhalte oder anderer Indikatoren wie der Beliebtheit mit Kanälen verknüpft. Mithilfe von Übersichtskategorien werden Kanäle so organisiert, dass YouTube-Nutzer die gesuchten Inhalte leichter finden können. Kanäle können zwar mit einer oder mehreren Guide-Kategorien verknüpft sein, aber wir können nicht garantieren, dass sie in Guide-Kategorien enthalten sind.
`i18nLanguage`	Bezeichnet eine App-Sprache, die von der YouTube-Website unterstützt wird. Die Anwendungssprache kann auch als UI-Sprache bezeichnet werden.
`i18nRegion`	Gibt ein geografisches Gebiet an, das ein YouTube-Nutzer als bevorzugte Inhaltsregion auswählen kann. Der Inhaltsbereich kann auch als Inhaltssprache bezeichnet werden.
`playlist`	Steht für eine einzelne YouTube-Playlist. Eine Playlist ist eine Sammlung von Videos, die nacheinander angesehen und mit anderen Nutzern geteilt werden können.
`playlistItem`	Kennzeichnet eine Ressource, die Teil einer Playlist ist, z. B. ein Video. Die Ressource playlistItem enthält außerdem Details dazu, wie die enthaltene

	Ressource in der Playlist verwendet wird.
`search result`	Enthält Informationen zu einem YouTube-Video, einem Kanal oder einer Playlist, die mit den in einer API-Anfrage angegebenen Suchparametern übereinstimmen. Obwohl ein Suchergebnis auf eine eindeutig identifizierbare Ressource wie ein Video verweist, hat es keine eigenen persistenten Daten.
`subscription`	Enthält Informationen zu einem YouTube-Nutzerabo. Ein Abo benachrichtigt Nutzer, wenn einem Kanal neue Videos hinzugefügt werden oder wenn ein anderer Nutzer eine von mehreren Aktionen auf YouTube ausführt, z. B. ein Video hochlädt, ein Video bewertet oder ein Video kommentiert.
`thumbnail`	Identifiziert Miniaturansichten, die einer Ressource zugeordnet sind.
`video`	Bezeichnet ein einzelnes YouTube-Video.
`videoCategory`	Kennzeichnet eine Kategorie, die mit hochgeladenen Videos verknüpft wurde oder werden könnte.
`watermark`	Kennzeichnet ein Bild, das während der Wiedergabe der Videos eines bestimmten Kanals angezeigt wird. Der Kanalinhaber kann auch einen Zielkanal angeben, zu dem das Bild verlinkt, sowie Zeitangaben, die bestimmen, wann das Wasserzeichen während der Videowiedergabe erscheint und wie lange es sichtbar ist.

Beachten Sie, dass eine Ressource in vielen Fällen Verweise auf andere Ressourcen enthält. Das Attribut `snippet.resourceId.videoId` einer `playlistItem`-Ressource identifiziert beispielsweise eine Videoressource, die wiederum vollständige Informationen über das Video enthält. Ein weiteres Beispiel: Ein Suchergebnis enthält entweder die Property `videoId`, `playlistId` oder `channelId`, die eine bestimmte Video-, Playlist- oder Kanalressource angibt.

Unterstützte Vorgänge

Die folgende Tabelle zeigt die gängigsten Methoden, die von der API unterstützt werden. Einige Ressourcen unterstützen auch andere Methoden, die für diese Ressourcen spezifischere Funktionen ausführen. Die Methode `videos.rate` verknüpft beispielsweise eine Nutzerbewertung mit einem Video und die Methode `thumbnails.set` lädt ein Video-Thumbnail auf YouTube hoch und verknüpft es mit einem Video.

Vorgänge	
`list`	Ruft eine Liste mit null oder mehr Ressourcen ab (`GET`).
`insert`	Erstellt (`POST`) eine neue Ressource.
`update`	Ändert (`PUT`) eine vorhandene Ressource, um Daten in Ihrer Anfrage widerzuspiegeln.
`delete`	Entfernt (`DELETE`) eine bestimmte Ressource.

Die API unterstützt derzeit Methoden zum Auflisten der unterstützten Ressourcentypen und Schreibvorgänge für viele Ressourcen.

In der folgenden Tabelle sind die Vorgänge aufgeführt, die für verschiedene Ressourcentypen unterstützt werden. Vorgänge zum Einfügen, Aktualisieren oder Löschen von Ressourcen erfordern immer eine Nutzerautorisierung (https://developers.google.com/youtube/v3/guides/authentication?hl=de). In einigen Fällen unterstützen `list`-Methoden sowohl autorisierte als auch nicht autorisierte Anfragen, bei denen nicht autorisierte Anfragen nur öffentliche Daten abrufen, während autorisierte Anfragen auch Informationen über den aktuell authentifizierten Nutzer oder private Anfragen abrufen können.

Unterstützte Vorgänge

	list	insert	update	delete
`activity`	✓	⊘	⊘	⊘
`caption`	✓	✓	✓	✓
`channel`	✓	⊘	⊘	⊘
`channelBanner`	⊘	✓	⊘	⊘
`channelSection`	✓	✓	✓	✓
`comment`	✓	✓	✓	✓
`commentThread`	✓	✓	✓	⊘
`guideCategory`	⊘	⊘	⊘	⊘
`i18nLanguage`	✓	⊘	⊘	⊘
`i18nRegion`	✓	⊘	⊘	⊘
`playlist`	✓	✓	✓	✓
`playlistItem`	✓	✓	✓	✓
`search result`	✓	⊘	⊘	⊘
`subscription`	✓	⊘	⊘	⊘
`thumbnail`	⊘	⊘	⊘	⊘
`video`	✓	✓	✓	✓
`videoCategory`	✓	⊘	⊘	⊘
`watermark`	⊘	⊘	⊘	⊘

Kontingentnutzung

Die YouTube Data API verwendet ein Kontingent, um sicherzustellen, dass Entwickler den Dienst wie vorgesehen verwenden und keine Anwendungen erstellen, die die Dienstqualität auf unfaire Weise verringern oder den Zugriff für andere einschränken. Für alle API-Anfragen, einschließlich ungültiger Anfragen, fallen mindestens ein Punktkontingent an. Das für Ihre Anwendung verfügbare Kontingent finden Sie in der API Console (https://console.cloud.google.com/?hl=de).

Projekte, bei denen die YouTube Data API aktiviert wird, haben eine standardmäßige Kontingentzuteilung von 10.000 Einheiten pro Tag – ein Betrag, der für die überwiegende Mehrheit unserer API-Nutzer ausreicht. Das Standardkontingent kann sich ändern und hilft uns dabei, die Kontingentzuweisung zu optimieren und unsere Infrastruktur auf eine Weise zu skalieren, die für unsere API-Nutzer sinnvoller ist. Sie können Ihre Kontingentnutzung in der API-Konsole auf der Seite Kontingente (https://console.developers.google.com/iam-admin/quotas?hl=de) einsehen.

Hinweis:Wenn Sie das Kontingentlimit erreichen, können Sie bis zum Abschluss der Kontingenterweiterung Antragsformular (https://support.google.com/youtube/contact/yt_api_form?hl=de) für die YouTube API-Dienste an.

Kontingentnutzung berechnen

Google berechnet Ihre Kontingentnutzung, indem Google jeder Anfrage Kosten zuweist. Verschiedene Arten von Vorgänge haben unterschiedliche Kontingentkosten. Beispiel:

- Eine Leseoperation, die eine Liste von Ressourcen – Kanälen, Videos, Playlists – abruft, in der Regel kostet 1 Einheit.
- Ein Schreibvorgang, der eine Ressource erstellt, aktualisiert oder löscht, verursacht in der Regel Kosten `50` Einheiten.
- Eine Suchanfrage kostet `100` Einheiten.
- Ein Video-Upload kostet `1600` Einheiten.

In der Tabelle Kontingentkosten für API-Anfragen (https://developers.google.com/youtube/v3/determine_quota_cost?hl=de) sehen Sie Kontingentkosten für jede API-Methode. Unter Berücksichtigung dieser Regeln können Sie die Anzahl der Anfragen abschätzen, die Ihre Anwendung pro Tag senden kann, ohne Ihr Kontingent zu überschreiten.

Teilressourcen

Die API ermöglicht und erfordert das Abrufen von Teilressourcen, sodass die Anwendungen das Übertragen, Parsen und Speichern nicht benötigter Daten vermeiden. Dieser Ansatz sorgt auch dafür, dass die API Netzwerk-, CPU- und Arbeitsspeicherressourcen effizienter verwendet.

Die API unterstützt zwei Anfrageparameter, die in den folgenden Abschnitten erläutert werden und mit denen Sie die Ressourceneigenschaften identifizieren können, die in API-Antworten enthalten sein sollen.

- Der Parameter `part` (#part) gibt Gruppen von Attributen an, die für eine Ressource zurückgegeben werden sollen.
- Der Parameter `fields` (#fields) filtert die API-Antwort so, dass nur bestimmte Attribute innerhalb der angeforderten Ressourcenteile zurückgegeben werden.

Verwendung des `part`-Parameters

Der Parameter `part` ist ein erforderlicher Parameter für jede API-Anfrage, die eine Ressource abruft oder zurückgibt. Der Parameter gibt mindestens ein (nicht verschachteltes) übergeordnetes Ressourcenattribut an, das in einer API-Antwort enthalten sein sollte. Eine Ressource vom Typ `video` (https://developers.google.com/youtube/v3/docs/videos?hl=de#resource) besteht beispielsweise aus folgenden Teilen:

```
snippet

contentDetails

fileDetails

player

processingDetails

recordingDetails

statistics

status

suggestions

topicDetails
```

All diese Teile sind Objekte, die verschachtelte Eigenschaften enthalten. Sie können sich diese Objekte als Gruppen von Metadatenfeldern vorstellen, die der API-Server abrufen könnte (oder nicht). Daher müssen Sie für den Parameter `part` die Ressourcenkomponenten auswählen, die Ihre Anwendung tatsächlich verwendet. Diese Anforderung dient hauptsächlich zwei Zwecken:

- Es verringert die Latenz, da der API-Server keine Zeit für das Abrufen von Metadatenfeldern benötigt, die nicht von Ihrer Anwendung verwendet werden.

- Sie reduziert die Bandbreitennutzung, indem die Menge unnötiger Daten, die Ihre Anwendung möglicherweise abrufen könnte, reduziert (oder beseitigt) wird.

Wenn im Laufe der Zeit Ressourcen hinzugefügt werden, werden diese Vorteile nur zunehmen, da Ihre Anwendung keine neu eingeführten Properties anfordert, die nicht unterstützt werden.

Verwendung des `fields`-Parameters

Der Parameter `fields` filtert die API-Antwort, die nur die im Parameterwert `part` identifizierten Ressourcenteile enthält, sodass die Antwort nur eine bestimmte Gruppe von Feldern enthält. Mit dem Parameter `fields` können Sie verschachtelte Attribute aus einer API-Antwort entfernen und so die Bandbreitennutzung weiter reduzieren. Der Parameter `part` kann nicht verwendet werden, um verschachtelte Eigenschaften aus einer Antwort zu filtern.

Die folgenden Regeln erklären die unterstützte Syntax für den Parameterwert `fields`, die grob auf der XPath-Syntax basiert:

- Verwende eine durch Kommas getrennte Liste (`fields=a,b`), um mehrere Felder auszuwählen.
- Verwenden Sie ein Sternchen (`fields=*`) als Platzhalter, um alle Felder anzugeben.
- Verwenden Sie Klammern (`fields=a(b,c)`), um eine Gruppe verschachtelter Attribute anzugeben, die in der API-Antwort enthalten sein sollen.
- Verwenden Sie einen Schrägstrich (`fields=a/b`), um eine verschachtelte Property anzugeben.

In der Praxis erlauben diese Regeln häufig, dass mehrere verschiedene `fields`-Parameterwerte dieselbe API-Antwort abrufen. Wenn Sie beispielsweise die ID, den Titel und die Position aller Elemente in einer Playlist abrufen möchten, können Sie einen der folgenden Werte verwenden:

- `fields=items/id,playlistItems/snippet/title,playlistItems/snippet/position`
- `fields=items(id,snippet/title,snippet/position)`
- `fields=items(id,snippet(title,position))`

Hinweis: Wie bei allen Abfrageparameterwerten muss der Wert des Parameters `fields` URL-codiert sein. Die Beispiele in diesem Dokument enthalten für eine bessere Lesbarkeit keine Codierung.

Beispiele für Teilanfragen

Die folgenden Beispiele veranschaulichen, wie Sie mit den Parametern `part` und `fields` dafür sorgen können, dass API-Antworten nur die Daten enthalten, die von Ihrer Anwendung verwendet werden:

1. Beispiel 1 gibt eine Videoressource zurück, die vier Teile sowie die Eigenschaften **`kind`** und **`etag`** enthält.
2. In Beispiel 2 wird eine Videoressource zurückgegeben, die zwei Teile sowie die Properties **`kind`** und **`etag`** enthält.
3. In Beispiel 3 wird eine Videoressource zurückgegeben, die zwei Teile enthält, aber die Properties **`kind`** und **`etag`** ausschließt.
4. Beispiel 4 gibt eine Videoressource zurück, die zwei Teile enthält, aber **`kind`** und **`etag`** sowie einige verschachtelte Eigenschaften im **`snippet`**-Objekt der Ressource ausschließt.

<ph type="x-smartling-placeholder">

Beispiel 1Beispiel 2 (#beispiel-2)Beispiel 3 (#beispiel-3)Beispiel 4 (#beispiel-4)
(#beispiel-1)

</ph>

URL: https://www.googleapis.com/youtube/v3/videos?id=7lCDEYXw3mM&key=YOUR_AP
&part=snippet,contentDetails,statistics,status

Description: This example retrieves a video resource and identifies several resource parts that should be included in the API response.

API response:


```
{
 "kind": "youtube#videoListResponse",
 "etag": "\"UCBpFjp2h75_b92t44sqraUcyu0/sDAlsG9NGKfr6v5AlPZKSEZdtqA\"",
 "videos": [
  {
   "id": "7lCDEYXw3mM",
   "kind": "youtube#video",
   "etag": "\"UCBpFjp2h75_b92t44sqraUcyu0/iYynQR8AtacsFUwWmrVaw4Smb_Q\"",
   "snippet": {
    "publishedAt": "2012-06-20T22:45:24.000Z",
    "channelId": "UC_x5XG1OV2P6uZZ5FSM9Ttw",
    "title": "Google I/O 101: Q&A On Using Google APIs",
    "description": "Antonio Fuentes speaks to us and takes questions on work:
    "thumbnails": {
     "default": {
      "url": "https://i.ytimg.com/vi/7lCDEYXw3mM/default.jpg"
     },
     "medium": {
      "url": "https://i.ytimg.com/vi/7lCDEYXw3mM/mqdefault.jpg"
     },
     "high": {
      "url": "https://i.ytimg.com/vi/7lCDEYXw3mM/hqdefault.jpg"
     }
    },
```

```
     "categoryId": "28"
    },
    "contentDetails": {
     "duration": "PT15M51S",
     "aspectRatio": "RATIO_16_9"
    },
    "statistics": {
     "viewCount": "3057",
     "likeCount": "25",
     "dislikeCount": "0",
     "favoriteCount": "17",
     "commentCount": "12"
    },
    "status": {
     "uploadStatus": "STATUS_PROCESSED",
     "privacyStatus": "PRIVACY_PUBLIC"
    }
   }
  ]
}
```


Leistungsoptimierung

ETags verwenden

Mit ETags, einem Standardteil des HTTP-Protokolls (https://www.rfc-editor.org/rfc/rfc9110.html#name-etag), können Anwendungen auf eine bestimmte Version einer bestimmten API-Ressource verweisen. Bei der Ressource kann es sich um einen ganzen Feed oder einen Artikel in diesem Feed handeln. Diese Funktion unterstützt folgende Anwendungsfälle:

- **Caching und bedingter Abruf** – Ihre Anwendung kann API-Ressourcen und ihre ETags im Cache speichern. Wenn Ihre Anwendung dann eine gespeicherte Ressource erneut anfordert, gibt sie das ETag an, das dieser Ressource zugeordnet ist. Wenn die Ressource geändert wurde, gibt die API die geänderte Ressource und das mit dieser Version der Ressource verknüpfte ETag zurück. Wenn die Ressource nicht geändert wurde, gibt die API eine HTTP 304-Antwort (`Not Modified`) zurück, die darauf hinweist, dass die Ressource nicht geändert wurde. Ihre Anwendung kann die Latenz und Bandbreitennutzung reduzieren, indem sie im Cache gespeicherte Ressourcen auf diese Weise bereitstellt.

  Die Clientbibliotheken für Google APIs unterstützen ETags unterschiedlich. Beispielsweise unterstützt die JavaScript-Clientbibliothek ETags über eine weiße Liste für zulässige Anfrageheader, die `If-Match` und `If-None-Match` enthält. Die Zulassungsliste ermöglicht ein normales Browser-Caching. Wenn sich das ETag einer Ressource nicht geändert hat,

kann die Ressource aus dem Browser-Cache bereitgestellt werden. Der Obj-C-Client hingegen unterstützt keine ETags.

- **Schutz vor versehentlichem Überschreiben von Änderungen**: ETags sorgen dafür, dass nicht versehentlich mehrere API-Clients gegenseitig überschreiben. Beim Aktualisieren oder Löschen einer Ressource kann Ihre Anwendung das ETag der Ressource angeben. Wenn der ETag nicht mit der neuesten Version dieser Ressource übereinstimmt, schlägt die API-Anfrage fehl.

Die Verwendung von ETags in Ihrer Anwendung bietet mehrere Vorteile:

- Die API reagiert schneller auf Anfragen nach im Cache gespeicherten, aber unveränderten Ressourcen, was zu einer geringeren Latenz und einer geringeren Bandbreitennutzung führt.
- Ihre Anwendung überschreibt nicht versehentlich Änderungen an einer Ressource, die von einem anderen API-Client vorgenommen wurden.

Google APIs Client Library for JavaScript (https://developers.google.com/api-client-library/javascript/start/start-js?hl=de) unterstützt `If-Match`- und `If-None-Match`-HTTP-Anfrageheader, wodurch ETags im Kontext des normalen Browser-Cachings funktionieren.

gzip verwenden

Sie können auch die Bandbreite reduzieren, die für jede API-Antwort benötigt wird, indem Sie die gzip-Komprimierung aktivieren. Für die Dekomprimierung von API-Antworten benötigt Ihre Anwendung zwar zusätzliche CPU-Zeit, doch in der Regel überwiegt der Vorteil der geringeren Netzwerkressourcen diese Kosten.

Führen Sie zwei Schritte aus, um eine mit gzip codierte Antwort zu erhalten:

- Legen Sie den `Accept-Encoding`-HTTP-Anfrageheader auf `gzip` fest.
- Ändern Sie Ihren User-Agent so, dass er den String `gzip` enthält.

Die folgenden HTTP-Beispielheader zeigen diese Anforderungen zum Aktivieren der gzip-Komprimierung:

```
Accept-Encoding: gzip
User-Agent: my program (gzip)
```

Websiterichtlinien von Google Developers (https://developers.google.com/site-policies?hl=de). Java ist eine eingetragene Marke von Oracle und/oder seinen Partnern.

Zuletzt aktualisiert: 2024-09-08 (UTC).