Package: youtubeR (via r-universe)

November 18, 2024

Title An Interface to the YouTube API

Version 0.0.0.9001

Description Manage YouTube videos from R.

License MIT + file LICENSE

URL `https://kevin-m-kent.github.io/youtubeR/`,
`https://github.com/kevin-m-kent/youtubeR`

BugReports `https://github.com/kevin-m-kent/youtubeR/issues`

Depends R (>= 4.1)

Imports cli, curl, fs, glue, httr2 (>= 1.0.0), jsonlite, purrr, rlang, snakecase

Suggests covr, httptest2, testthat (>= 3.0.0), withr

Config/testthat/edition 3

Encoding UTF-8

Language en-US

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Config/pak/sysreqs make libicu-dev libssl-dev

Repository https://jonthegeek.r-universe.dev

RemoteUrl https://github.com/kevin-m-kent/youtubeR

RemoteRef HEAD

RemoteSha 3b3fd5e43e1d4dc7108b5df39691ea28d9e1f044

Contents

browse_gc_credentials *Visit Google Cloud Credentials page*

Description

Launch the Google Cloud "Credentials" page (where you can configure OAuth 2.0 Clients). After you configure a client, copy/paste the Client ID and Client Secret into the `YOUTUBE_CLIENT_ID` and `YOUTUBE_CLIENT_SECRET` environment variables. We recommend placing these environment variables in your `.Renviron` file.

Usage

```
browse_gc_credentials()
```

Value

The url of the "Credentials" page, invisibly.

Examples

```
gc_credentials_url <- browse_gc_credentials()
# Copy/paste values from your client.
Sys.setenv(YOUTUBE_CLIENT_ID = "12345-ab12c.apps.googleusercontent.com")
Sys.setenv(YOUTUBE_CLIENT_SECRET = "ABCD-eFg_H")
```

`datetime` *Datetime vectors*

Description

Create an object of type "POSIXct". This function is used to provide a length-0 placeholder value in function definitions.

Usage

```
datetime()
```

Value

A length-0 POSIXct object.

Examples

```
datetime()
```

`get_my_channel_detail_playlist_ids` *Get my channel detail playlist ids*

Description

Get ids of playlists that contain details about the content of a channel.

Usage

```
get_my_channel_detail_playlist_ids(
  client = yt_construct_client(),
  cache_disk = getOption(”yt_cache_disk”, FALSE),
  cache_key = getOption(”yt_cache_key”, NULL),
  token = NULL
)
```

Arguments

`client` A YouTube OAuth client created with `yt_construct_client()`.

`cache_disk` Should the access token be cached on disk? This reduces the number of times that you need to re-authenticate at the cost of storing access credentials on disk. Learn more in `https://httr2.r-lib.org/articles/oauth.html`.

`cache_key` If you are authenticating with multiple users using the same client, use this key to differentiate between those users.

`token` A YouTube API OAuth token, or the `access_token` string from such a token. We recommend that you instead supply a `client`, in which case an appropriate token will be located if possible.

Value

The *contentDetails* object encapsulates information about the channel's content.

Property	Class	Description
watchHistory	character	The ID of the playlist that contains the channel's watch history. Use the playlistItems.insert and pl
likes	character	The ID of the playlist that contains the channel's liked videos. Use the playlistItems.insert and pla
favorites	character	The ID of the playlist that contains the channel's favorite videos. Use the playlistItems.insert and
watchLater	character	The ID of the playlist that contains the channel's watch later playlist. Use the playlistItems.insert
uploads	character	The ID of the playlist that contains the channel's uploaded videos. Use the videos.insert method to

Examples

```
get_my_channel_details()
```

---

get_playlist_items *Get playlist items*

Description

Retrieve a list of items in a given playlist.

Usage

```
get_playlist_items(
  playlist_id,
  max_results = 100,
  client = yt_construct_client(),
  cache_disk = getOption("youtubeR.cache_disk", FALSE),
  cache_key = getOption("youtubeR.cache_key", NULL),
  token = NULL
)
```

Arguments

`playlist_id` The ID of a playlist.

`max_results` The maximum number of results to return.

`client` A YouTube OAuth client created with `yt_construct_client()`.

`cache_disk` Should the access token be cached on disk? This reduces the number of times that you need to re-authenticate at the cost of storing access credentials on disk. Learn more in `https://httr2.r-lib.org/articles/oauth.html`.

`cache_key` If you are authenticating with multiple users using the same client, use this key to differentiate between those users.

`token` A YouTube API OAuth token, or the `access_token` string from such a token. We recommend that you instead supply a `client`, in which case an appropriate token will be located if possible.

Value

A list of videos, each of which has a videoId and a videoPublishedAt.

Examples

```
get_playlist_items(playlist_id = get_upload_playlist_id())
```

---

get_playlist_video_ids

Get playlist video ids

Description

Retrieves video ids for a particular playlist.

Usage

```
get_playlist_video_ids(
  playlist_id,
  max_results = 100,
  client = yt_construct_client(),
  cache_disk = getOption("yt_cache_disk", FALSE),
  cache_key = getOption("yt_cache_key", NULL),
  token = NULL
)
```

Arguments

`playlist_id` The ID of a playlist.

`max_results` The maximum number of results to return.

`client` A YouTube OAuth client created with `yt_construct_client()`.

`cache_disk` Should the access token be cached on disk? This reduces the number of times that you need to re-authenticate at the cost of storing access credentials on disk. Learn more in `https://httr2.r-lib.org/articles/oauth.html`.

`cache_key` If you are authenticating with multiple users using the same client, use this key to differentiate between those users.

`token` A YouTube API OAuth token, or the `access_token` string from such a token. We recommend that you instead supply a `client`, in which case an appropriate token will be located if possible.

Value

A character vector of video ids.

Examples

```
get_playlist_video_ids(playlist_id = get_upload_playlist_id())
```

`get_upload_playlist_id`

Get Playlist Ids

Description

Retrieves the upload playlist ids.

Usage

```
get_upload_playlist_id(
  client = yt_construct_client(),
  cache_disk = getOption("yt_cache_disk", FALSE),
  cache_key = getOption("yt_cache_key", NULL),
  token = NULL
)
```

Arguments

`client` A YouTube OAuth client created with `yt_construct_client()`.

`cache_disk` Should the access token be cached on disk? This reduces the number of times that you need to re-authenticate at the cost of storing access credentials on disk.

Learn more in `https://httr2.r-lib.org/articles/oauth.html`.

`cache_key` If you are authenticating with multiple users using the same client, use this key to differentiate between those users.

`token` A YouTube API OAuth token, or the `access_token` string from such a token. We recommend that you instead supply a `client`, in which case an appropriate token will be located if possible.

Value

list of playlist ids for uploads

Examples

```
get_upload_playlist_id()
```

get_video_processing_details

Get video processing details

Description

Retrieves the processing details for a vector of video IDs

Usage

```
get_video_processing_details(
  video_ids,
  client = yt_construct_client(),
  cache_disk = getOption("yt_cache_disk", FALSE),
  cache_key = getOption("yt_cache_key", NULL),
  token = NULL
)
```

Arguments

video_ids	A character vector of video ids.
client	A YouTube OAuth client created with yt_construct_client().
cache_disk	Should the access token be cached on disk? This reduces the number of times that you need to re-authenticate at the cost of storing access credentials on disk. Learn more in https://httr2.r-lib.org/articles/oauth.html.
cache_key	If you are authenticating with multiple users using the same client, use this key to differentiate between those users.
token	A YouTube API OAuth token, or the access_token string from such a token. We recommend that you instead supply a client, in which case an appropriate token will be located if possible.

Value

A list of processing status details. processingStatus is probably of most interest

Examples

```
get_upload_playlist_id() |>
  get_playlist_items() |>
  get_video_processing_details()
```

yt_authenticate *Authenticate with a YouTube OAuth client*

Description

Load or generate a YouTube OAuth token for use in the other functions in this package. The primary use of this function is to cache values early in a script (so the user can walk away). Otherwise the other functions in this package will prompt for authentication when needed. Once the values are cached, the rest of this package will use them by default for that client.

Usage

```
yt_authenticate(
  client = yt_construct_client(),
  force = FALSE,
  refresh_token = NULL
)
```

Arguments

`client`	A YouTube OAuth client created with `yt_construct_client()`.
`force`	A logical indicating whether to force a refresh of the token.
`refresh_token`	A refresh token associated with this `client`. This parameter exists primarily for testing. If you wish to provide a refresh token (for example, for automated processes), we recommend setting a `YOUTUBE_REFRESH_TOKEN` environment variable.

Value

A YouTube OAuth token, invisibly.

Examples

```
token <- yt_authenticate()
```

yt_call_api *Call the YouTube API*

Description

If you use this function, please open an issue in the youtubeR GitHub repository describing your use case. We intend to wrap all endpoints such that this function will not be necessary for users.

Usage

```
yt_call_api(
  endpoint,
  query = NULL,
  body = NULL,
  method = NULL,
  client = yt_construct_client(),
  cache_disk = getOption("youtuberR.cache_disk", FALSE),
  cache_key = getOption("youtuberR.cache_key", NULL),
  token = NULL,
  base_url = c("basic", "upload", "resumable_upload")
)
```

Arguments

`endpoint` The path to an endpoint. Optionally, a list with the path plus variables to `glue::glue()` into the path.

`query` An optional list of parameters to pass in the query portion of the request.

`body` An optional list of parameters to pass in the body portion of the request.

`method` If the method is something other than GET or POST, supply it. Case is ignored.

`client` A YouTube OAuth client created with `yt_construct_client()`.

`cache_disk` Should the access token be cached on disk? This reduces the number of times that you need to re-authenticate at the cost of storing access credentials on disk. Learn more in `https://httr2.r-lib.org/articles/oauth.html`.

`cache_key` If you are authenticating with multiple users using the same client, use this key to differentiate between those users.

`token` A YouTube API OAuth token, or the `access_token` string from such a token. We recommend that you instead supply a `client`, in which case an appropriate token will be located if possible.

`base_url` Which family of URLs to use. Almost everything will use the default basic URL.

Value

The result of the call.

`yt_construct_client` *Construct a YouTube OAuth client*

Description

Builds the OAuth client object for google apis.

Usage

```
yt_construct_client(
  client_id = Sys.getenv("YOUTUBE_CLIENT_ID"),
  client_secret = Sys.getenv("YOUTUBE_CLIENT_SECRET")
)
```

Arguments

`client_id` A Google Cloud OAuth 2.0 client ID. We recommend you save it as an environment variable, `YOUTUBE_CLIENT_ID`.

`client_secret` A Google Cloud OAuth 2.0 client secret. We recommend you save it as an environment variable, `YOUTUBE_CLIENT_SECRET`.

Value

An `httr2::oauth_client()` object.

Examples

```
client <- yt_construct_client()
```

`yt_has_client_envvars` *Check for YouTube client environment variables*

Description

Reports whether YouTube client environment variables are set (or, if arguments are provided, check that they have characters).

Usage

```
yt_has_client_envvars(
  client_id = Sys.getenv("YOUTUBE_CLIENT_ID"),
  client_secret = Sys.getenv("YOUTUBE_CLIENT_SECRET")
)
```

Arguments

`client_id` A Google Cloud OAuth 2.0 client ID. We recommend you save it as an environment variable, `YOUTUBE_CLIENT_ID`.

`client_secret` A Google Cloud OAuth 2.0 client secret. We recommend you save it as an environment variable, `YOUTUBE_CLIENT_SECRET`.

Value

A logical indicating whether the variables are available.

Examples

```
yt_has_client_envvars()
yt_has_client_envvars("an_id_string", "a_secret_string")
```

`yt_schema_video_localization` *Video localization*

Description

Localized versions of certain video properties (e.g. title).

Usage

```
yt_schema_video_localization(description = character(), title = character())
```

Arguments

`description`	(character scalar) Localized version of the video's description.
`title`	(character scalar) Localized version of the video's title.

Value

A list with any non-zero-length properties.

Examples

```
yt_schema_video_localization()
yt_schema_video_localization(description = "A description.")
```

`yt_schema_video_snippet` *Video snippet*

Description

Basic details about a video, including title, description, uploader, thumbnails and category.

Usage

```
yt_schema_video_snippet(
  title = character(),
  description = character(),
  tags = character(),
  category_id = character(),
  default_language = character()
)
```

Arguments

`title`	(character scalar) The video's title.
`description`	(character scalar) The video's description.
`tags`	(character vector) A list of keyword tags associated with the video. Tags may contain spaces.
`category_id`	(character scalar) The YouTube video category associated with the video.
`default_language`	(character scalar) The language of the videos's default snippet.

Value

A list with any non-zero-length properties.

Examples

```
yt_schema_video_snippet()
yt_schema_video_snippet(title = "An example video", tags = c("a", "b"))
```

`yt_schema_video_status` *Video status*

Description

The status object contains information about the video's uploading, processing, and privacy statuses.

Usage

```
yt_schema_video_status(
  embeddable = logical(),
  license = character(),
  privacy_status = character(),
  public_stats_viewable = logical(),
  publish_at = datetime(),
  self_declared_made_for_kids = logical()
)
```

Arguments

`embeddable`	(logical) This value indicates if the video can be embedded on another website.
`license`	(factor scalar) The video's license. values: youtube, creativeCommon
`privacy_status`	(factor scalar) The video's privacy status. values: public, unlisted, private
`public_stats_viewable`	(logical) This value indicates if the extended video statistics on the watch page can be viewed by everyone. Note that the view count, likes, etc will still be visible if this is disabled.

`publish_at` (datetime scalar) The date and time when the video is scheduled to publish. It can be set only if the privacy status of the video is private.

`self_declared_made_for_kids` (logical) Whether the video is made for kids.

Value

A list with any non-zero-length properties.

Examples

```
yt_schema_video_status()
yt_schema_video_status(embeddable = TRUE)
```

`yt_videos_delete` *Delete video*

Description

Deletes a resource.

Usage

```
yt_videos_delete(
  video_id,
  client = yt_construct_client(),
  cache_disk = getOption("youtuberR.cache_disk", FALSE),
  cache_key = getOption("youtuberR.cache_key", NULL),
  token = NULL
)
```

Arguments

`video_id` (character scalar) The id parameter specifies the YouTube video ID for the resource that is being deleted. In a video resource, the id property specifies the video's ID.

`client` A YouTube OAuth client created with `yt_construct_client()`.

`cache_disk` Should the access token be cached on disk? This reduces the number of times that you need to re-authenticate at the cost of storing access credentials on disk. Learn more in `https://httr2.r-lib.org/articles/oauth.html`.

`cache_key` If you are authenticating with multiple users using the same client, use this key to differentiate between those users.

`token` A YouTube API OAuth token, or the `access_token` string from such a token. We recommend that you instead supply a `client`, in which case an appropriate token will be located if possible.

Value

The id of the deleted video.

`yt_videos_insert` *Insert video*

Description

Inserts a new resource into this collection.

Usage

```
yt_videos_insert(
  video_path,
  snippet = yt_schema_video_snippet(),
  localizations = list(),
  status = yt_schema_video_status(),
  recording_date = datetime(),
  client = yt_construct_client(),
  cache_disk = getOption("youtuberR.cache_disk", FALSE),
  cache_key = getOption("youtuberR.cache_key", NULL),
  token = NULL
)
```

Arguments

`video_path`	(character scalar) Path to a video file to upload.
`snippet`	Basic details about a video, including title, description, uploader, thumbnails and category. See `yt_schema_video_snippet()` for details.
`localizations`	A list of named localizations objects. The localizations object contains localized versions of the basic details about the video, such as its title and description. The name should be a valid language key, which is a BCP-47 language code. See `yt_schema_video_localization()` for details on the format for each localization.
`status`	Basic details about a video category, such as its localized title. See `yt_schema_video_status()` for details.
`recording_date`	(datetime scalar) The date and time when the video was recorded.
`client`	A YouTube OAuth client created with `yt_construct_client()`.
`cache_disk`	Should the access token be cached on disk? This reduces the number of times that you need to re-authenticate at the cost of storing access credentials on disk. Learn more in `https://httr2.r-lib.org/articles/oauth.html`.
`cache_key`	If you are authenticating with multiple users using the same client, use this key to differentiate between those users.
`token`	A YouTube API OAuth token, or the `access_token` string from such a token. We recommend that you instead supply a `client`, in which case an appropriate token will be located if possible.

Value

The id of the uploaded video.

`yt_videos_update` *Update video*

Description

Updates an existing resource.

Usage

```
yt_videos_update(
  video_id,
  snippet = yt_schema_video_snippet(),
  localizations = list(),
  status = yt_schema_video_status(),
  recording_date = datetime(),
  client = yt_construct_client(),
  cache_disk = getOption("youtuberR.cache_disk", FALSE),
  cache_key = getOption("youtuberR.cache_key", NULL),
  token = NULL
)
```

Arguments

`video_id` (character scalar) The id parameter specifies the YouTube video ID for the resource that is being updated In a video resource, the id property specifies the video's ID.

`snippet` Basic details about a video, including title, description, uploader, thumbnails and category. See `yt_schema_video_snippet()` for details.

`localizations` A list of named localizations objects. The localizations object contains localized versions of the basic details about the video, such as its title and description. The name should be a valid language key, which is a BCP-47 language code. See `yt_schema_video_localization()` for details on the format for each localization.

`status` Basic details about a video category, such as its localized title. See `yt_schema_video_status()` for details.

`recording_date` (datetime scalar) The date and time when the video was recorded.

`client` A YouTube OAuth client created with `yt_construct_client()`.

`cache_disk` Should the access token be cached on disk? This reduces the number of times that you need to re-authenticate at the cost of storing access credentials on disk. Learn more in `https://httr2.r-lib.org/articles/oauth.html`.

`cache_key` If you are authenticating with multiple users using the same client, use this key to differentiate between those users.

`token` A YouTube API OAuth token, or the `access_token` string from such a token. We recommend that you instead supply a `client`, in which case an appropriate token will be located if possible.

Value

The id of the updated video.

Index