Package: rytstat (via r-universe)

October 18, 2026

Title Work with 'YouTube API'

Version 0.3.2

Description Provide function for get data from 'YouTube Data API' <https://developers.google.com/youtube/v3/docs/>, 'YouTube Analytics API' <https://developers.google.com/youtube/analytics/reference/> and 'YouTube Reporting API' <https://developers.google.com/youtube/reporting/v1/reports>.

Imports cli, dplyr, gargle, snakecase, stringr, tidyr, pbapply, httr, withr, rlang

URL https://selesnow.github.io/rytstat/docs/

BugReports https://github.com/selesnow/rytstat/issues

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.0

Config/pak/sysreqs cmake make libicu-dev libuv1-dev libssl-dev

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

Date/Publication 2024-05-30 12:10:34 UTC

RemoteUrl https://github.com/selesnow/rytstat

RemoteRef HEAD

RemoteSha a825a11ceee8436293d2691cbb38f159765e4bed

Contents

rytstat-package 2
ryt_auth 3
ryt_auth_configure 4
ryt_deauth 6

ryt_get_analytics 7
ryt_get_channel_activities 8
ryt_get_channels 12
ryt_get_comments 15
ryt_get_playlist_items 16
ryt_get_playlists 18
ryt_get_report_types 20
ryt_get_video_details 23
ryt_get_videos 29
ryt_has_token 29
ryt_search 30
ryt_token 31
ryt_user 32

Index **33**

`rytstat-package` *rytstat: Work with 'YouTube API'*

Description

Provide function for get data from 'YouTube Data API' https://developers.google.com/youtube/v3/docs/, 'YouTube Analytics API' https://developers.google.com/youtube/analytics/reference/ and 'YouTube Reporting API' https://developers.google.com/youtube/reporting/v1/reports.

Author(s)

Maintainer: Alexey Seleznev <selesnow@gmail.com> (ORCID)

Other contributors:

- Netpeak [copyright holder]

See Also

Useful links:

- https://selesnow.github.io/rytstat/docs/
- Report bugs at https://github.com/selesnow/rytstat/issues

ryt_auth *Authorization in YouTube API*

Description

Authorize rytstat to view and manage your YouTube Account. This function is a wrapper around `gargle::token_fetch()`.

By default, you are directed to a web browser, asked to sign in to your Google account, and to grant rytstat permission to operate on your behalf with YouTube. By default, with your permission, these user credentials are cached in a folder below your home directory, from where they can be automatically refreshed, as necessary. Storage at the user level means the same token can be used across multiple projects and tokens are less likely to be synced to the cloud by accident.

If you are interacting with R within a browser (applies to RStudio Server, RStudio Workbench, and RStudio Cloud), you need a variant of this flow, known as out-of-band auth ("oob"). If this does not happen automatically, you can request it yourself with `use_oob = TRUE` or, more persistently, by setting an option via `options(gargle_oob_default = TRUE)`.

Usage

```
ryt_auth(
  email = gargle::gargle_oauth_email(),
  path = NULL,
  cache = gargle::gargle_oauth_cache(),
  use_oob = gargle::gargle_oob_default(),
  token = NULL
)
```

Arguments

`email`	Optional. Allows user to target a specific Google identity.
`path`	Path to JSON file with identifying the service account
`cache`	Specifies the OAuth token cache.
`use_oob`	Whether to prefer "out of band" authentication.
`token`	A token with class Token2.0 or an object of

Details

Most users, most of the time, do not need to call `ryt_auth()` explicitly – it is triggered by the first action that requires authorization. Even when called, the default arguments often suffice. However, when necessary, this function allows the user to explicitly:

- Declare which Google identity to use, via an email address. If there are multiple cached tokens, this can clarify which one to use. It can also force rytstat to switch from one identity to another. If there's no cached token for the email, this triggers a return to the browser to choose the identity and give consent. You can specify just the domain by using a glob pattern. This means that a script containing `email = "*@example.com"` can be run without further tweaks on the machine of either `alice@example.com` or `bob@example.com`.

- Use a service account token or workload identity federation.
- Bring their own Token2.0.
- Specify non-default behavior re: token caching and out-of-bound authentication.
- Customize scopes.

For details on the many ways to find a token, see `gargle::token_fetch()`. For deeper control over auth, use `ryt_auth_configure()` to bring your own OAuth app or API key. Read more about gargle options, see gargle::gargle_options.

Value

Token2.0

See Also

Other auth functions: `ryt_auth_configure()`, `ryt_deauth()`

Examples

```
## Not run:
## load/refresh existing credentials, if available
## otherwise, go to browser for authentication and authorization
ryt_auth()

## force use of a token associated with a specific email
ryt_auth(email = "yourname@example.com")

## force a menu where you can choose from existing tokens or
## choose to get a new one
ryt_auth(email = NA)

## -----------------------
## use own OAuth client app
ryt_auth_configure(
    path = "path/to/your/oauth_client.json"
)

ryt_auth(email = "yourname@example.com")

## End(Not run)
```

`ryt_auth_configure` *Edit and view auth configuration*

Description

These functions give more control over and visibility into the auth configuration than `ryt_auth()` does. `ryt_auth_configure()` lets the user specify their own:

- OAuth app, which is used when obtaining a user token.
- API key. If rytstat is de-authorized via `ryt_deauth()`, all requests are sent with an API key in lieu of a token. See the vignette How to get your own API credentials for more. If the user does not configure these settings, internal defaults are used. `ryt_oauth_app()` and `ryt_api_key()` retrieve the currently configured OAuth app and API key, respectively.

Usage

```
ryt_auth_configure(app, path, api_key)

ryt_auth_cache_path()

ryt_open_auth_cache_folder()

ryt_api_key()

ryt_oauth_app()
```

Arguments

app	OAuth app, in the sense of `httr::oauth_app()`.
path	JSON downloaded from Google Cloud Platform Console, containing a client id (aka key) and secret, in one of the forms supported for the `txt` argument of `jsonlite::fromJSON()` (typically, a file path or JSON string).
api_key	API key.

Value

- `ryt_auth_configure()`: An object of R6 class gargle::AuthState, invisibly.
- `ryt_oauth_app()`: the current user-configured `httr::oauth_app()`.
- `ryt_api_key()`: the current user-configured API key.

See Also

Other auth functions: `ryt_auth()`, `ryt_deauth()`

Examples

```
## Not run:
# see and store the current user-configured OAuth app (probaby `NULL`)
(original_app <- ryt_oauth_app())

# see and store the current user-configured API key (probaby `NULL`)
(original_api_key <- ryt_api_key())
```

```
if (require(httr)) {
  # bring your own app via client id (aka key) and secret
  google_app <- httr::oauth_app(
    "my-awesome-google-api-wrapping-package",
    key = "YOUR_CLIENT_ID_GOES_HERE",
    secret = "YOUR_SECRET_GOES_HERE"
  )
  google_key <- "YOUR_API_KEY"
  ryt_auth_configure(app = google_app, api_key = google_key)

  # confirm the changes
  ryt_oauth_app()
  ryt_api_key()

  # bring your own app via JSON downloaded from Google Developers Console
  # this file has the same structure as the JSON from Google
  ryt_auth_configure(path = app_path)

  # confirm the changes
  ryt_oauth_app()

}

# restore original auth config
gs4_auth_configure(app = original_app, api_key = original_api_key)

## End(Not run)
```

`ryt_deauth` *Suspend authorization*

Description

Put rytstat into a de-authorized state. Instead of sending a token, rytstat will send an API key. This can be used to access public resources for which no Google sign-in is required. This is handy for using rytstat in a non-interactive setting to make requests that do not require a token. It will prevent the attempt to obtain a token interactively in the browser. The user can configure their own API key via `ryt_auth_configure()` and retrieve that key via `ryt_api_key()`. In the absence of a user-configured key, a built-in default key is used.

Usage

```
ryt_deauth()
```

Value

only suspend authorization

See Also

Other auth functions: `ryt_auth_configure()`, `ryt_auth()`

`ryt_get_analytics` *Get statistics from 'YouTube Analytics API'*

Description

The YouTube Analytics API enables you to generate custom reports containing YouTube Analytics data. The API supports reports for channels and for content owners.

Usage

```
ryt_get_analytics(
  start_date = Sys.Date() - 14,
  end_date = Sys.Date(),
  metrics = c("views", "estimatedMinutesWatched", "averageViewDuration",
    "averageViewPercentage", "subscribersGained"),
  dimensions = "day",
  filters = NULL,
  sort = NULL,
  max_results = NULL,
  start_index = NULL
)
```

Arguments

`start_date`	The start date for fetching YouTube Analytics data. The value should be in YYYY-MM-DD format.
`end_date`	The end date for fetching YouTube Analytics data. The value should be in YYYY-MM-DD format.
`metrics`	Character vector of YouTube Analytics metrics, such as views or likes,dislikes. See the documentation for channel reports or a list of the reports that you can retrieve and the metrics available in each report. The Metrics document contains definitions for all of the metrics.
`dimensions`	Character vector of YouTube Analytics dimensions, such as video or ageGroup,gender. The Dimensions document contains definitions for all of the dimensions.
`filters`	Character vector of filters that should be applied when retrieving YouTube Analytics data. The documentation for channel reports reports identifies the dimensions that can be used to filter each report, and the Dimensions document defines those dimensions.
`sort`	Character vector of dimensions or metrics that determine the sort order for YouTube Analytics data. By default the sort order is ascending. The - prefix causes descending sort order.
`max_results`	The maximum number of rows to include in the response.
`start_index`	The 1-based index of the first entity to retrieve. (The default value is 1.) Use this parameter as a pagination mechanism along with the max_results parameter.

Value

tibble with analytics data

Examples

```
## Not run:
# auth
ryt_auth()

# get list of your videos
videos <- ryt_get_videos()

# function for loading video stat
get_videos_stat <- function(video_id) {

  data <- ryt_get_analytics(
    metrics = c('views', 'likes', 'dislikes', 'comments', 'shares'),
    filters = stringr::str_glue('video=={video_id}')
  )

  if ( nrow(data) > 0 ) {
    data <- mutate(data, video_id = video_id)
}
}

# load video stat
video_stat <- purrr::map_df(videos$id_video_id, get_videos_stat)

# join stat with video metadata
video_stat <- left_join(video_stat,
                        videos,
                        by = c("video_id" = "id_video_id")) %>%
              select(video_id,
                     title,
                     day,
                     views,
                     likes,
                     dislikes,
                     comments,
                     shares)

## End(Not run)
```

---

ryt_get_channel_activities

Returns a list of channel activity events

Description

Returns a list of channel activity events

Usage

```
ryt_get_channel_activities(
  part = c("contentDetails", "id", "snippet"),
  fields = NULL
)
```

Arguments

part	The part parameter specifies a comma-separated list of one or more activity resource properties that the API response will include. See API documentation.
fields	Fields of video metadata see Properties

Details

Parts and fields **part:**

- contentDetails
- id
- snippet

parts and fields details

- kind - Identifies the API resource's type.
- etag - The Etag of this resource.
- id - The ID that YouTube uses to uniquely identify the activity.
- snippet - The snippet object contains basic details about the activity, including the activity's type and group ID.
  - snippet.publishedAt - The date and time that the activity occurred.
  - snippet.channelId - The ID that YouTube uses to uniquely identify the channel associated with the activity.
  - snippet.title - The title of the resource primarily associated with the activity.
  - snippet.description - The description of the resource primarily associated with the activity.
  - snippet.thumbnails - A map of thumbnail images associated with the resource that is primarily associated with the activity.
  - snippet.thumbnails.(key) - Valid key values are: default, medium, high, standard, maxres
  - snippet.thumbnails.(key).url - The image's URL.
  - snippet.thumbnails.(key).width - The image's width.
  - snippet.thumbnails.(key).height - The image's height.
  - snippet.channelTitle - Channel title for the channel responsible for this activity
  - snippet.type - The type of activity that the resource describes.
  - snippet.groupId - The group ID associated with the activity.
- contentDetails - The contentDetails object contains information about the content associated with the activity.
  - contentDetails.upload - The upload object contains information about the uploaded video.

- contentDetails.upload.videoId - The ID that YouTube uses to uniquely identify the uploaded video.
- contentDetails.like - The like object contains information about a resource that received a positive (like) rating.
- contentDetails.like.resourceId - The resourceId object contains information that identifies the rated resource.
- contentDetails.like.resourceId.kind - The type of the API resource.
- contentDetails.like.resourceId.videoId - The ID that YouTube uses to uniquely identify the video, if the rated resource is a video.
- contentDetails.favorite - The favorite object contains information about a video that was marked as a favorite video.
- contentDetails.favorite.resourceId - The resourceId object contains information that identifies the resource that was marked as a favorite.
- contentDetails.favorite.resourceId.kind - The type of the API resource.
- contentDetails.favorite.resourceId.videoId - The ID that YouTube uses to uniquely identify the favorite video.
- contentDetails.comment - The comment object contains information about a resource that received a comment. This property is only present if the snippet.type is comment.
- contentDetails.comment.resourceId - The resourceId object contains information that identifies the resource associated with the comment.
- contentDetails.comment.resourceId.kind - The type of the API resource.
- contentDetails.comment.resourceId.videoId - The ID that YouTube uses to uniquely identify the video associated with a comment.
- contentDetails.comment.resourceId.channelId - The ID that YouTube uses to uniquely identify the channel associated with a comment.
- contentDetails.subscription - The subscription object contains information about a channel that a user subscribed to.
- contentDetails.subscription.resourceId - The resourceId object contains information that identifies the resource that the user subscribed to.
- contentDetails.subscription.resourceId.kind - The type of the API resource.
- contentDetails.subscription.resourceId.channelId - The ID that YouTube uses to uniquely identify the channel that the user subscribed to.
- contentDetails.playlistItem - The playlistItem object contains information about a new playlist item.
- contentDetails.playlistItem.resourceId - The resourceId object contains information that identifies the resource that was added to the playlist.
- contentDetails.playlistItem.resourceId.kind - The type of the API resource.
- contentDetails.playlistItem.resourceId.videoId - The ID that YouTube uses to uniquely identify the video that was added to the playlist. This property is only present if the resourceId.kind is youtube#video.
- contentDetails.playlistItem.playlistId - The value that YouTube uses to uniquely identify the playlist.
- contentDetails.playlistItem.playlistItemId - The value that YouTube uses to uniquely identify the item in the playlist.

- contentDetails.recommendation - The recommendation object contains information about a recommended resource. This property is only present if the snippet.type is recommendation.
- contentDetails.recommendation.resourceId - The resourceId object contains information that identifies the recommended resource.
- contentDetails.recommendation.resourceId.kind - The type of the API resource.
- contentDetails.recommendation.resourceId.videoId - The ID that YouTube uses to uniquely identify the video, if the recommended resource is a video.
- contentDetails.recommendation.resourceId.channelId - The ID that YouTube uses to uniquely identify the channel, if the recommended resource is a channel.
- contentDetails.recommendation.reason - The reason that the resource is recommended to the user.
- contentDetails.recommendation.seedResourceId - The seedResourceId object contains information about the resource that caused the recommendation.
- contentDetails.recommendation.seedResourceId.kind - The type of the API resource.
- contentDetails.recommendation.seedResourceId.videoId - The ID that YouTube uses to uniquely identify the video, if the recommendation was caused by a particular video.
- contentDetails.recommendation.seedResourceId.channelId - The ID that YouTube uses to uniquely identify the channel, if the recommendation was caused by a particular channel.
- contentDetails.recommendation.seedResourceId.playlistId - The ID that YouTube uses to uniquely identify the playlist, if the recommendation was caused by a particular playlist.
- contentDetails.social - The social object contains details about a social network post.
- contentDetails.social.type - The name of the social network.
- contentDetails.social.resourceId - The resourceId object encapsulates information that identifies the resource associated with a social network post.
- contentDetails.social.resourceId.kind - The type of the API resource.
- contentDetails.social.resourceId.videoId - The ID that YouTube uses to uniquely identify the video featured in a social network post, if the post refers to a video.
- contentDetails.social.resourceId.channelId - The ID that YouTube uses to uniquely identify the channel featured in a social network post, if the post refers to a channel. This property will only be present if the value of the social.
- contentDetails.social.resourceId.playlistId - The ID that YouTube uses to uniquely identify the playlist featured in a social network post, if the post refers to a playlist.
- contentDetails.social.author - The author of the social network post.
- contentDetails.social.referenceUrl - The URL of the social network post.
- contentDetails.social.imageUrl - An image of the post's author.
- contentDetails.channelItem - The channelItem object contains details about a resource that was added to a channel.
- contentDetails.channelItem.resourceId - The resourceId object contains information that identifies the resource that was added to the channel.

Value

tibble with channel activies

Examples

```
## Not run:

channel_activities <- ryt_get_channel_activities()


## End(Not run)
```

`ryt_get_channels` *Get channel info from 'YouTube API'*

Description

Get channel info from 'YouTube API'

Usage

```
ryt_get_channels(
 part = c("contentDetails", "id", "snippet", "statistics", "status", "topicDetails"),
  fields = NULL
)
```

Arguments

`part`	The part parameter specifies a comma-separated list of one or more channel resource properties that the API response will include., see API documentation.
`fields`	Allows you to specify individual nested part fields

Details

Parts and fields **part:**

- auditDetails
- brandingSettings
- contentDetails
- contentOwnerDetails
- id
- localizations
- snippet
- statistics
- status
- topicDetails

parts and fields details

- kind - Identifies the API resource's type.
- etag - The Etag of this resource.
- id - The ID that YouTube uses to uniquely identify the channel.
- snippet - The snippet object contains basic details about the channel, such as its title, description, and thumbnail images.
  - snippet.title - The channel's title.
  - snippet.description - The channel's description. The property's value has a maximum length of 1000 characters.
  - snippet.customUrl - The channel's custom URL. The YouTube Help Center explains eligibility requirements for getting a custom URL as well as how to set up the URL.
  - snippet.publishedAt - The date and time that the channel was created. The value is specified in ISO 8601 format.
  - snippet.thumbnails - A map of thumbnail images associated with the channel. For each object in the map, the key is the name of the thumbnail image, and the value is an object that contains other information about the thumbnail.
  - snippet.thumbnails.(key) - Valid key values are: default, medium, high.
  - snippet.thumbnails.(key).url - The image's URL. See the snippet.thumbnails property definition for additional guidelines on using thumbnail URLs in your application.
  - snippet.thumbnails.(key).width - The image's width.
  - snippet.thumbnails.(key).height - The image's height.
  - snippet.defaultLanguage - The language of the text in the channel resource's snippet.title and snippet.description properties.
  - snippet.localized - The snippet.localized object contains a localized title and description for the channel or it contains the channel's title and description in the default language for the channel's metadata.
  - snippet.localized.title - The localized channel title.
  - snippet.localized.description - The localized channel description.
  - snippet.country - The country with which the channel is associated.
- contentDetails - The contentDetails object encapsulates information about the channel's content.
  - contentDetails.relatedPlaylists - The relatedPlaylists object is a map that identifies playlists associated with the channel, such as the channel's uploaded videos or liked videos.
  - contentDetails.relatedPlaylists.likes - The ID of the playlist that contains the channel's liked videos.
  - contentDetails.relatedPlaylists.uploads - The ID of the playlist that contains the channel's uploaded videos.
- statistics - The statistics object encapsulates statistics for the channel.
  - statistics.viewCount - The number of times the channel has been viewed.
  - statistics.subscriberCount - The number of subscribers that the channel has.
  - statistics.hiddenSubscriberCount - Indicates whether the channel's subscriber count is publicly visible.
  - statistics.videoCount - The number of public videos uploaded to the channel.
- topicDetails - The topicDetails object encapsulates information about topics associated with the channel.

  - topicDetails.topicIds[] - A list of topic IDs associated with the channel.
  - topicDetails.topicCategories[] - A list of Wikipedia URLs that describe the channel's content.
- status - The status object encapsulates information about the privacy status of the channel.
  - status.privacyStatus - Privacy status of the channel.
  - status.isLinked - Indicates whether the channel data identifies a user that is already linked to either a YouTube username or a Google+ account.
  - status.longUploadsStatus - Indicates whether the channel is eligible to upload videos that are more than 15 minutes long.
  - status.madeForKids - This value indicates whether the channel is designated as child-directed, and it contains the current "made for kids" status of the channel.
  - status.selfDeclaredMadeForKids - In a channels.update request, this property allows the channel owner to designate the channel as child-directed.
- brandingSettings - The brandingSettings object encapsulates information about the branding of the channel.
  - brandingSettings.channel - The channel object encapsulates branding properties of the channel page.
  - brandingSettings.channel.title - The channel's title. The title has a maximum length of 30 characters.
  - brandingSettings.channel.description - The channel description, which appears in the channel information box on your channel page. The property's value has a maximum length of 1000 characters.
  - brandingSettings.channel.keywords - Keywords associated with your channel.
  - brandingSettings.channel.trackingAnalyticsAccountId - The ID for a Google Analytics account that you want to use to track and measure traffic to your channel.
  - brandingSettings.channel.moderateComments - This setting determines whether user-submitted comments left on the channel page need to be approved by the channel owner to be publicly visible.
  - brandingSettings.channel.unsubscribedTrailer - The video that should play in the featured video module in the channel page's browse view for unsubscribed viewers.
  - brandingSettings.channel.defaultLanguage - The language of the text in the channel resource's snippet.title and snippet.description properties.
  - brandingSettings.channel.country - The country with which the channel is associated. Update this property to set the value of the snippet.country property.
  - brandingSettings.watch - The watch object encapsulates branding properties of the watch pages for the channel's videos.
  - brandingSettings.watch.textColor - The text color for the video watch page's branded area.
  - brandingSettings.watch.backgroundColor - The background color for the video watch page's branded area.
- auditDetails - The auditDetails object encapsulates channel data that a multichannel network (MCN) would evaluate while determining whether to accept or reject a particular channel.
  - auditDetails.overallGoodStanding - This field indicates whether there are any issues with the channel. Currently, this field represents the result of the logical AND operation over

the communityGuidelinesGoodStanding, copyrightStrikesGoodStanding, and contentIdClaimsGoodStanding properties, meaning that this property has a value of true if all of those other properties also have a value of true.
  - auditDetails.communityGuidelinesGoodStanding - Indicates whether the channel respects YouTube's community guidelines.
  - auditDetails.copyrightStrikesGoodStanding - Indicates whether the channel has any copyright strikes.
  - auditDetails.contentIdClaimsGoodStanding - Indicates whether the channel has any unresolved claims.
- contentOwnerDetails - The contentOwnerDetails object encapsulates channel data that is relevant for YouTube Partners linked with the channel.
  - contentOwnerDetails.contentOwner - The ID of the content owner linked to the channel.
  - contentOwnerDetails.timeLinked - The date and time of when the channel was linked to the content owner. The value is specified in ISO 8601 format.
- localizations - The localizations object encapsulates translations of the channel's metadata.
  - localizations.(key) - The language of the localized metadata associated with the key value.
  - localizations.(key).title - The localized channel title.
  - localizations.(key).description - The localized channel description.

Value

tibble with channel metadata

Examples

```
## Not run:
channels <- ryt_get_channels()


## End(Not run)
```

`ryt_get_comments` *Returns a list of comment threads of video or channel*

Description

Returns a list of comment threads of video or channel

Usage

```
ryt_get_comments(
  video_id = NULL,
  channel_id = NULL,
  text_format = c("plainText", "html")
)
```

Arguments

video_id	YouTUbe Video ID
channel_id	YouTube Channel ID
text_format	Set this parameter's value to html or plainText to instruct the API to return the comments left by users in html formatted or in plain text. The default value is plainText

Value

tibble with comments

See Also

Reporting API Documentation.

Examples

```
## Not run:
# all comments
comments <- ryt_get_comments()

# videos comments
video_comments <- ryt_get_comments(video_id = 'fW7gGS^G78')

## End(Not run)
```

---

ryt_get_playlist_items

Get playlist items data on 'YouTube'

Description

Get playlist items data on 'YouTube'

Usage

```
ryt_get_playlist_items(
  playlist_id,
  part = c("contentDetails", "id", "snippet", "status"),
  fields = NULL,
  cl = NULL
)
```

Arguments

playlist_id	Playlist ID, see ryt_get_playlists.
part	The part parameter specifies a comma-separated list of one or more playlistItem resource properties that the API response will include. see API documentation.
fields	The fields parameter filters the API response, which only contains the resource parts identified in the part parameter value, so that the response only includes a specific set of fields. see API documentation.
cl	A cluster object created by makeCluster, or an integer to indicate number of child-processes (integer values are ignored on Windows) for parallel evaluations (see Details on performance).

Details

Parts and fields **part:**

- contentDetails
- id
- snippet
- status

parts and fields details

- kind - Identifies the API resource's type. The value will be youtube#playlistItemListResponse.
- etag - The Etag of this resource.
- id - The ID that YouTube uses to uniquely identify the playlist item.
- snippet - The snippet object contains basic details about the playlist item, such as its title and position in the playlist.
  - snippet/publishedAt - The date and time that the item was added to the playlist.
  - snippet/channelId - The ID that YouTube uses to uniquely identify the user that added the item to the playlist.
  - snippet/title - The item's title.
  - snippet/description - The item's description.
  - snippet/thumbnails - A map of thumbnail images associated with the playlist item.
  - snippet/thumbnails/(key) - Valid key values are: default, medium, high, standard, maxres
  - snippet/thumbnails/(key)/url - The image's URL.
  - snippet/thumbnails/(key)/width - The image's width.
  - snippet/thumbnails/(key)/height - The image's height.
  - snippet/channelTitle - The channel title of the channel that the playlist item belongs to.
  - snippet/videoOwnerChannelTitle - The channel title of the channel that uploaded this video.
  - snippet/videoOwnerChannelId - The channel ID of the channel that uploaded this video.
  - snippet/playlistId - The ID that YouTube uses to uniquely identify the playlist that the playlist item is in.
  - snippet/position - The order in which the item appears in the playlist.

  - snippet/resourceId - The id object contains information that can be used to uniquely identify the resource that is included in the playlist as the playlist item.
  - snippet/resourceId.kind - The kind, or type, of the referred resource.
  - snippet/resourceId/videoId - If the snippet.resourceId.kind property's value is youtube#video, then this property will be present and its value will contain the ID that YouTube uses to uniquely identify the video in the playlist.
- contentDetails - The contentDetails object is included in the resource if the included item is a YouTube video. The object contains additional information about the video.
  - contentDetails/videoId - The ID that YouTube uses to uniquely identify a video.
  - contentDetails/note - A user-generated note for this item. The property value has a maximum length of 280 characters.
  - contentDetails/videoPublishedAt - The date and time that the video was published to YouTube.
- status - The status object contains information about the playlist item's privacy status.
  - status/privacyStatus - The playlist item's privacy status.

Value

tibble with playlist items details

Examples

```
## Not run:
# get playlist ids and title
pl <- ryt_get_playlists(part = c('id', 'snippet'), fields = 'items(id, snippet/title)')

# get itemms of first playlist
pli <- ryt_get_playlist_items(
    playlist_id = pl$id[1],
    part = c('contentDetails', 'snippet'),
    fields = 'items(id,snippet/channelId,snippet/title,contentDetails/videoId)'
 )

## End(Not run)
```

`ryt_get_playlists` *Get playlist from 'YouTube API'*

Description

Get playlist from 'YouTube API'

Usage

```
ryt_get_playlists(
 part = c("contentDetails", "id", "localizations", "player", "snippet", "status"),
  fields = NULL
)
```

Arguments

`part`	parts of playist metadata, see API documentation.
`fields`	Fields of video metadata, see API documentation.

Details

Parts and fields **part:**

- contentDetails
- id
- localizations
- player
- snippet
- status

parts and fields details

- kind - Identifies the API resource's type.
- etag - The Etag of this resource.
- id - The ID that YouTube uses to uniquely identify the playlist.
- snippet - The snippet object contains basic details about the playlist, such as its title and description.
  - snippet/publishedAt - The date and time that the playlist was created.
  - snippet/channelId - The ID that YouTube uses to uniquely identify the channel that published the playlist.
  - snippet/title - The playlist's title.
  - snippet/description - The playlist's description.
  - snippet/thumbnails - A map of thumbnail images associated with the playlist.
  - snippet/thumbnails/(key) - Valid key values are: default, medium, high, standard, maxres
  - snippet/thumbnails/(key)/url - The image's URL.
  - snippet/thumbnails/(key)/width - The image's width.
  - snippet/thumbnails/(key)/height - The image's height.
  - snippet/channelTitle - The channel title of the channel that the video belongs to.
  - snippet/defaultLanguage - The language of the text in the playlist resource's snippet.title and snippet.description properties.
  - snippet/localized - The snippet.localized object contains either a localized title and description for the playlist or the title in the default language for the playlist's metadata.
  - snippet/localized/title - The localized playlist title.
  - snippet/localized/description - The localized playlist description.
- status - The status object contains status information for the playlist.
  - status/privacyStatus - The playlist's privacy status.
- contentDetails - The contentDetails object contains information about the playlist content, including the number of videos in the playlist.

  - contentDetails/itemCount - The number of videos in the playlist.
- player - The player object contains information that you would use to play the playlist in an embedded player.
  - player/embedHtml - An iframe tag that embeds a player that will play the playlist.
- localizations - The localizations object encapsulates translations of the playlist's metadata.
  - localizations/(key) - The language of the localized text associated with the key value.
  - localizations/(key)/title - The localized playlist title.
  - localizations/(key)/description - The localized playlist description.

Value

tibble with playlist metadata

Examples

```
## Not run:
pl <- ryt_get_playlists(
    part = c('id', 'contentDetails', 'snippet'),
    fields = 'items(id,snippet/channelId,snippet/title,contentDetails/itemCount)'
 )

## End(Not run)
```

---

ryt_get_report_types *Returns a list of report types that the channel or content owner can retrieve. Each item in the list contains an id property, which identifies the report's ID, and you need this value to schedule a reporting job.*

Description

By creating a reporting job, you are instructing YouTube to generate that report on a daily basis. The report is available within 24 hours of the time that the job is created.

Each resource in the response contains an id property, which specifies the ID that YouTube uses to uniquely identify the job. You need that ID to retrieve the list of reports that have been generated for the job or to delete the job.

Usage

```
ryt_get_report_types()

ryt_create_job(
 report_type = c("channel_annotations_a1", "channel_basic_a2", "channel_cards_a1",
    "channel_combined_a2", "channel_demographics_a1", "channel_device_os_a2",
  "channel_end_screens_a1", "channel_playback_location_a2", "channel_province_a2",
  "channel_sharing_service_a1", "channel_playback_location_a2", "channel_province_a2",
  "channel_sharing_service_a1", "channel_subtitles_a2", "channel_traffic_source_a2",
```

```
   "playlist_basic_a1", "playlist_device_os_a1", "playlist_playback_location_a1",
    "playlist_province_a1", "playlist_traffic_source_a1")
)

ryt_get_job_list()

ryt_get_report_list(
  job_id,
  created_after = NULL,
  start_time_at_or_after = NULL,
  start_time_before = NULL
)

ryt_get_report(download_url)

ryt_get_report_metadata(job_id, report_id)

ryt_delete_job(job_id)
```

Arguments

report_type	The type of report that the job creates. The property value corresponds to the id of a reportType as retrieved from the ryt_get_report_types function.
job_id	The ID that YouTube uses to uniquely identify the job that is being deleted. Use ryt_get_job_list.
created_after	If specified, this parameter indicates that the API response should only contain reports created after the specified date and time, including new reports with backfilled data. Note that the value pertains to the time that the report is created and not the dates associated with the returned data. The value is a timestamp in RFC3339 UTC "Zulu" format, accurate to microseconds. Example: "2015-10-02T15:01:23.045678Z".
start_time_at_or_after	This parameter indicates that the API response should only contain reports if the earliest data in the report is on or after the specified date. Whereas the createdAfter parameter value pertains to the time the report was created, this date pertains to the data in the report. The value is a timestamp in RFC3339 UTC "Zulu" format, accurate to microseconds. Example: "2015-10-02T15:01:23.045678Z".
start_time_before	This parameter indicates that the API response should only contain reports if the earliest data in the report is before the specified date. Whereas the createdAfter parameter value pertains to the time the report was created, this date pertains to the data in the report. The value is a timestamp in RFC3339 UTC "Zulu" format, accurate to microseconds. Example: "2015-10-02T15:01:23.045678Z".
download_url	download URL, you can get it by ryt_get_report_list or ryt_get_report_metadata
report_id	The ID that YouTube uses to uniquely identify the report that is being retrieved. Use ryt_get_report_list

Value

ryt_get_report_types: tibble with report types

ryt_reports_create_job: No return value, called for side effects

ryt_get_job_list: tibble with jobs metadata

ryt_get_report_list: tibble with reports metadata

ryt_get_report: tibble with report data

ryt_get_report_metadata: list with report metadata

ryt_reports_delete_job: No return value, called for side effects

See Also

Reporting API Documentation: Method reportTypes.list.

Reporting API Documentation: Method jobs.create

Reporting API Documentation: Method jobs.list

Reporting API Documentation: Method jobs.reports.list

Reporting API Documentation: Data Model

Reporting API Documentation: Method jobs.reports.get

Reporting API Documentation: Method jobs.delete

Examples

```
## Not run:
 # auth
 ryt_auth('me@gmail.com')

 # get reporting data
 ## create job
 ryt_reports_create_job('channel_basic_a2')

 ## get job list
 jobs <- ryt_get_job_list()

 ## get job report list
 reports <- ryt_get_report_list(
   job_id = jobs$id[1],
   created_after = '2021-10-20T15:01:23.045678Z'
 )

 ## get report data
 data <- ryt_get_report(
   download_url = reports$downloadUrl[1]
 )

 ## delete job
 ryt_reports_delete_job(jobs$id[1])

## End(Not run)
```

`ryt_get_video_details` *Get detail data of your videos on 'YouTube'*

Description

Get detail data of your videos on 'YouTube'

Usage

```
ryt_get_video_details(
  video_id,
  part = c("contentDetails", "fileDetails", "id", "liveStreamingDetails",
   "localizations", "player", "processingDetails", "recordingDetails", "snippet",
    "statistics", "status", "suggestions", "topicDetails"),
  fields = NULL,
  cl = NULL
)
```

Arguments

`video_id`	Video ID, see `ryt_get_videos`.
`part`	The parameter identifies one or more top-level (non-nested) resource properties that should be included in an API response. Posible values: snippet, contentDetails, fileDetails, player, processingDetails, recordingDetails, statistics, status, suggestions, topicDetails.See API documentation.
`fields`	Fields of video metadata, see API documentation.
`cl`	A cluster object created by `makeCluster`, or an integer to indicate number of child-processes (integer values are ignored on Windows) for parallel evaluations (see Details on performance).

Details

For get more information about videos parts and fields go this link

Properties:

- kind - Identifies the API resource's type. The value will be youtube#video.
- etag - The Etag of this resource.
- id - The ID that YouTube uses to uniquely identify the video.
- snippet - The snippet object contains basic details about the video, such as its title, description, and category.
  - snippet/publishedAt - The date and time that the video was published. Note that this time might be different than the time that the video was uploaded.
  - snippet/channelId - The ID that YouTube uses to uniquely identify the channel that the video was uploaded to.
  - snippet/title - The video's title.

  - snippet/description - The video's description.
  - snippet/thumbnails - A map of thumbnail images associated with the video.
  - snippet/thumbnails/(key) - Valid key values are: default, medium, high, standard, maxres
  - snippet/thumbnails/(key)/url - The image's URL.
  - snippet/thumbnails/(key)/width - The image's width.
  - snippet/thumbnails/(key)/height - The image's height.
  - snippet/channelTitle - Channel title for the channel that the video belongs to.
  - snippet/tags[] - A list of keyword tags associated with the video.
  - snippet/categoryId - The YouTube video category associated with the video.
  - snippet/liveBroadcastContent - Indicates if the video is an upcoming/active live broadcast.
  - snippet/defaultLanguage - The language of the text in the video resource's snippet.title and snippet.description properties.
  - snippet/localized - The snippet.localized object contains either a localized title and description for the video or the title in the default language for the video's metadata.
  - snippet/localized/title - The localized video title.
  - snippet/localized/description - The localized video description.
  - snippet/defaultAudioLanguage - The default_audio_language property specifies the language spoken in the video's default audio track.
- contentDetails - The contentDetails object contains information about the video content, including the length of the video and an indication of whether captions are available for the video.
  - contentDetails/duration - The length of the video.
  - contentDetails/dimension - Indicates whether the video is available in 3D or in 2D.
  - contentDetails/definition - Indicates whether the video is available in high definition (HD) or only in standard definition.
  - contentDetails/caption - Indicates whether captions are available for the video.
  - contentDetails/licensedContent - Indicates whether the video represents licensed content, which means that the content was uploaded to a channel linked to a YouTube content partner and then claimed by that partner.
  - contentDetails/regionRestriction - The regionRestriction object contains information about the countries where a video is (or is not) viewable.
  - contentDetails/regionRestriction/allowed[] - A list of region codes that identify countries where the video is viewable.
  - contentDetails/regionRestriction/blocked[] - A list of region codes that identify countries where the video is blocked.
  - contentDetails/contentRating - Specifies the ratings that the video received under various rating schemes.
  - contentDetails/contentRating/acbRating - The video's Australian Classification Board (ACB) or Australian Communications and Media Authority (ACMA) rating. ACMA ratings are used to classify children's television programming.
  - contentDetails/contentRating/agcomRating - The video's rating from Italy's Autorità per le Garanzie nelle Comunicazioni (AGCOM).
  - contentDetails/contentRating/anatelRating - The video's Anatel (Asociación Nacional de Televisión) rating for Chilean television.

  - contentDetails/contentRating/bbfcRating - The video's British Board of Film Classification (BBFC) rating.
  - contentDetails/contentRating/bfvcRating - The video's rating from Thailand's Board of Film and Video Censors.
  - contentDetails/contentRating/bmukkRating - The video's rating from the Austrian Board of Media Classification.
  - contentDetails/contentRating/catvRating - Rating system for Canadian TV - Canadian TV Classification System The video's rating from the Canadian Radio-Television and Telecommunications Commission (CRTC) for Canadian English-language broadcasts.
  - contentDetails/contentRating/catvfrRating - The video's rating from the Canadian Radio-Television and Telecommunications Commission (CRTC) for Canadian French-language broadcasts.
  - contentDetails/contentRating/cbfcRating - The video's Central Board of Film Certification (CBFC - India) rating.
  - contentDetails/projection - Specifies the projection format of the video.
  - contentDetails/hasCustomThumbnail - Indicates whether the video uploader has provided a custom thumbnail image for the video. This property is only visible to the video uploader.

- status - The status object contains information about the video's uploading, processing, and privacy statuses.
  - status/uploadStatus - The status of the uploaded video.
  - status/failureReason - This value explains why a video failed to upload. This property is only present if the uploadStatus property indicates that the upload failed.
  - status/rejectionReason - This value explains why YouTube rejected an uploaded video.
  - status/privacyStatus - The video's privacy status.
  - status/publishAt - 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.
  - status/license - The video's license.
  - status/embeddable - This value indicates whether the video can be embedded on another website.
  - status/publicStatsViewable - This value indicates whether the extended video statistics on the video's watch page are publicly viewable.
  - status/madeForKids - This value indicates whether the video is designated as child-directed, and it contains the current "made for kids" status of the video.
  - status/selfDeclaredMadeForKids - In a videos.insert or videos.update request, this property allows the channel owner to designate the video as being child-directed.
- statistics - The statistics object contains statistics about the video.
  - statistics/viewCount - The number of times the video has been viewed.
  - statistics/likeCount - The number of users who have indicated that they liked the video.
  - statistics/dislikeCount - The number of users who have indicated that they disliked the video.
  - statistics/commentCount - The number of comments for the video.
- player - The player object contains information that you would use to play the video in an embedded player.

  - player/embedHtml - An iframe tag that embeds a player that plays the video.
  - player/embedHeight - The height of the embedded player returned in the player.embedHtml property.
  - player/embedWidth - The width of the embedded player returned in the player.embedHtml property.
- topicDetails - The topicDetails object encapsulates information about topics associated with the video.
  - topicDetails/relevantTopicIds[] - A list of topic IDs that are relevant to the video.
  - topicDetails/topicCategories[] - A list of Wikipedia URLs that provide a high-level description of the video's content.
- recordingDetails - The recordingDetails object encapsulates information about the location, date and address where the video was recorded.
  - recordingDetails/location - The geolocation information associated with the video.
  - recordingDetails/recordingDate - The date and time when the video was recorded.
- fileDetails - The fileDetails object encapsulates information about the video file that was uploaded to YouTube, including the file's resolution, duration, audio and video codecs, stream bitrates, and more.
  - fileDetails/fileName - The uploaded file's name.
  - fileDetails/fileSize - The uploaded file's size in bytes.
  - fileDetails/fileType - The uploaded file's type as detected by YouTube's video processing engine.
  - fileDetails/container - The uploaded video file's container format.
  - fileDetails/videoStreams[] - A list of video streams contained in the uploaded video file.
  - fileDetails/videoStreams[]/widthPixels - The encoded video content's width in pixels.
  - fileDetails/videoStreams[]/heightPixels - The encoded video content's height in pixels.
  - fileDetails/videoStreams[]/frameRateFps - The video stream's frame rate, in frames per second.
  - fileDetails/videoStreams[]/aspectRatio - The video content's display aspect ratio, which specifies the aspect ratio in which the video should be displayed.
  - fileDetails/videoStreams[]/codec - The video codec that the stream uses.
  - fileDetails/videoStreams[]/bitrateBps - The video stream's bitrate, in bits per second.
  - fileDetails/videoStreams[]/rotation - The amount that YouTube needs to rotate the original source content to properly display the video.
  - fileDetails/videoStreams[]/vendor - A value that uniquely identifies a video vendor.
  - fileDetails/audioStreams[] - A list of audio streams contained in the uploaded video file.
  - fileDetails/audioStreams[]/channelCount - The number of audio channels that the stream contains.
  - fileDetails/audioStreams[]/codec - The audio codec that the stream uses.
  - fileDetails/audioStreams[]/bitrateBps - The audio stream's bitrate, in bits per second.
  - fileDetails/audioStreams[]/vendor - A value that uniquely identifies a video vendor.
  - fileDetails/durationMs - The length of the uploaded video in milliseconds.
  - fileDetails/bitrateBps - The uploaded video file's combined (video and audio) bitrate in bits per second.

  - fileDetails/creationTime - The date and time when the uploaded video file was created.
- processingDetails - The processingDetails object encapsulates information about YouTube's progress in processing the uploaded video file.
  - processingDetails/processingStatus - The video's processing status.
  - processingDetails/processingProgress - The processingProgress object contains information about the progress YouTube has made in processing the video.
  - processingDetails/processingProgress/partsTotal - An estimate of the total number of parts that need to be processed for the video.
  - processingDetails/processingProgress/partsProcessed - The number of parts of the video that YouTube has already processed.
  - processingDetails/processingProgress/timeLeftMs - An estimate of the amount of time, in millseconds, that YouTube needs to finish processing the video.
  - processingDetails/processingFailureReason - The reason that YouTube failed to process the video.
  - processingDetails/fileDetailsAvailability - This value indicates whether file details are available for the uploaded video.
  - processingDetails/processingIssuesAvailability - This value indicates whether the video processing engine has generated suggestions that might improve YouTube's ability to process the the video, warnings that explain video processing problems, or errors that cause video processing problems.
  - processingDetails/tagSuggestionsAvailability - This value indicates whether keyword (tag) suggestions are available for the video.
  - processingDetails/editorSuggestionsAvailability - This value indicates whether video editing suggestions, which might improve video quality or the playback experience, are available for the video.
  - processingDetails/thumbnailsAvailability - This value indicates whether thumbnail images have been generated for the video.
- suggestions - The suggestions object encapsulates suggestions that identify opportunities to improve the video quality or the metadata for the uploaded video.
  - suggestions/processingErrors[] - A list of errors that will prevent YouTube from successfully processing the uploaded video.
  - suggestions/processingWarnings[] - A list of reasons why YouTube may have difficulty transcoding the uploaded video or that might result in an erroneous transcoding.
  - suggestions/processingHints[] - A list of suggestions that may improve YouTube's ability to process the video.
  - suggestions/tagSuggestions[] - A list of keyword tags that could be added to the video's metadata to increase the likelihood that users will locate your video when searching or browsing on YouTube.
  - suggestions/tagSuggestions[]/tag - The keyword tag suggested for the video.
  - suggestions/tagSuggestions[].categoryRestricts[] - A set of video categories for which the tag is relevant.
  - suggestions/editorSuggestions[] - A list of video editing operations that might improve the video quality or playback experience of the uploaded video.
- liveStreamingDetails - The liveStreamingDetails object contains metadata about a live video broadcast.

  - liveStreamingDetails/actualStartTime - The time that the broadcast actually started.
  - liveStreamingDetails/actualEndTime - The time that the broadcast actually ended.
  - liveStreamingDetails/scheduledStartTime - The time that the broadcast is scheduled to begin.
  - liveStreamingDetails/scheduledEndTime - The time that the broadcast is scheduled to end.
  - liveStreamingDetails/concurrentViewers - The number of viewers currently watching the broadcast.
  - liveStreamingDetails/activeLiveChatId - The ID of the currently active live chat attached to this video.
- localizations - The localizations object contains translations of the video's metadata.
  - localizations/(key) - The language of the localized text associated with the key value.
  - localizations/(key).title - The localized video title.
  - localizations/(key).description - The localized video description.

Value

tibble with video details

Examples

```
## Not run:
# get all videos
videos <- ryt_get_videos()

# get all videos metadata
videos_details <- ryt_get_video_details(
    video_id = videos$id_video_id
)

# get only snippet and statistics part
videos_details <- ryt_get_video_details(
    video_id = videos$id_video_id,
    part = c('snippet', 'statistics')
)

# get only id, channelId, title and view_count fields
videos_details <- ryt_get_video_details(
    video_id = videos$id_video_id,
    part = c('snippet', 'statistics'),
    fields = "items(id,snippet(channelId,title),statistics(viewCount))"
)

# same with other fields syntax like part/field
# get only id, channelId, title and view_count fields
videos_details <- ryt_get_video_details(
    video_id = videos$id_video_id,
    part = c('snippet', 'statistics'),
    fields = "items(id,snippet/channelId,snippet/title,statistics/viewCount)"
)
```

```
## End(Not run)
```

`ryt_get_videos` *Get list of your videos from 'YouTube'*

Description

Get list of your videos from 'YouTube'

Usage

```
ryt_get_videos(fields = NULL)
```

Arguments

`fields`	Fields of video metadata, see API documentation.

Value

tibble with video list

`ryt_has_token` *Is there a token on hand?*

Description

Reports whether rytstat has stored a token, ready for use in downstream requests.

Usage

```
ryt_has_token()
```

Value

Logical.

See Also

Other low-level API functions: `ryt_token()`

`ryt_search` *Search channels, playlists or videos on YouTube*

Description

Returns a collection of search results that match the query parameters specified in the API request. By default, a search result set identifies matching video, channel, and playlist resources, but you can also configure queries to only retrieve a specific type of resource.

Usage

```
ryt_search(part = "snippet", ...)
```

Arguments

`part`	The part parameter specifies a comma-separated list of one or more search resource properties that the API response will include. Set the parameter value to snippet.
`...`	Filters and Optional parameters for resources search. See list of all allowed params. You can set params names in came or snake case.

Value

resources list

Examples

```
## Not run:
# search videos by query
search_res_videos <- ryt_search(
  type            = 'video',
  q               = 'r language tutorial',
  published_after = '2022-03-01T00:00:00Z',
  published_before = '2022-06-01T00:00:00Z',
  max_results     = 10
)

# search playlists by query
search_res_playlists <- ryt_search(
  type             = 'playlist',
  q                = 'r language tutorial',
  published_after  = '2022-03-01T00:00:00Z',
  published_before = '2022-06-01T00:00:00Z',
  max_results      = 50
)

# search channel by query
search_res_channels <- ryt_search(
type             = 'channel',
```

```
q                = 'r language tutorial',
published_after  = '2022-03-01T00:00:00Z',
  published_before = '2022-06-01T00:00:00Z',
  max_results      = 50
)

# Search in own videos
search_own_dplyr_videos <- ryt_search(
  type             = 'video',
  for_mine         = TRUE,
  q                = 'dplyr'
)

# Search channels and get title and id
search_chn <-  ryt_search(
   type   = 'channel',
   q      = 'R4marketing',
   fields = 'items(snippet(title,channelId))'
)

# Search videos in the channel by query and channel id
search_channel_dplyr_videos <- ryt_search(
  type       = 'video',
  q          = 'dplyr',
  channel_id = "UCyHC6R3mCCP8bhD9tPbjnzQ"
)

## End(Not run)
```

`ryt_token` *Produce configured token*

Description

For internal use or for those programming around the YouTube API. Returns a token pre-processed with `httr::config()`. Most users do not need to handle tokens "by hand" or, even if they need some control, `ryt_auth()` is what they need. If there is no current token, `ryt_auth()` is called to either load from cache or initiate OAuth2.0 flow. If auth has been deactivated via `ryt_deauth()`, `ryt_token()` returns `NULL`.

Usage

```
ryt_token()
```

Value

A `request` object (an S3 class provided by httr).

See Also

Other low-level API functions: `ryt_has_token()`

`ryt_user` *Get info on current user*

Description

Reveals the email address of the user associated with the current token. If no token has been loaded yet, this function does not initiate auth.

Usage

```
ryt_user()
```

Value

An email address or, if no token has been loaded, `NULL`.

See Also

`gargle::token_userinfo()`, `gargle::token_email()`, `gargle::token_tokeninfo()`

Index

∗ **auth functions**
    ryt_auth, 3
    ryt_auth_configure, 4
    ryt_deauth, 6
∗ **low-level API functions**
    ryt_has_token, 29
    ryt_token, 31
∗ **reporting api functions**
    ryt_get_report_types, 20

gargle::AuthState, *5*
gargle::gargle_options, *4*
gargle::token_email(), *32*
gargle::token_fetch(), *3, 4*
gargle::token_tokeninfo(), *32*
gargle::token_userinfo(), *32*

httr, *31*
httr::config(), *31*
httr::oauth_app(), *5*

jsonlite::fromJSON(), *5*

makeCluster, *17*, *23*

ryt_api_key *(*ryt_auth_configure*)*, 4
ryt_api_key(), *6*
ryt_auth, 3, *5*, *7*
ryt_auth(), *5*, *31*
ryt_auth_cache_path
        *(*ryt_auth_configure*)*, 4
ryt_auth_configure, 4, *4*, *7*
ryt_auth_configure(), *4*, *6*
ryt_create_job *(*ryt_get_report_types*)*,
        20
ryt_deauth, *4, 5*, 6
ryt_deauth(), *5*, *31*
ryt_delete_job *(*ryt_get_report_types*)*,
        20
ryt_get_analytics, 7
ryt_get_channel_activities, 8
ryt_get_channels, 12
ryt_get_comments, 15
ryt_get_job_list, *21*
ryt_get_job_list
        *(*ryt_get_report_types*)*, 20
ryt_get_playlist_items, 16
ryt_get_playlists, *17*, 18
ryt_get_report *(*ryt_get_report_types*)*,
        20
ryt_get_report_list, *21*
ryt_get_report_list
        *(*ryt_get_report_types*)*, 20
ryt_get_report_metadata, *21*
ryt_get_report_metadata
        *(*ryt_get_report_types*)*, 20
ryt_get_report_types, 20, *21*
ryt_get_video_details, 23
ryt_get_videos, *23*, 29
ryt_has_token, 29, *32*
ryt_oauth_app *(*ryt_auth_configure*)*, 4
ryt_open_auth_cache_folder
        *(*ryt_auth_configure*)*, 4
ryt_search, 30
ryt_token, *29*, 31
ryt_user, 32
rytstat *(*rytstat-package*)*, 2
rytstat-package, 2

Token2.0, *3, 4*