

# Package ‘tubern’

May 8, 2026

**Title** R Client for the YouTube Analytics and Reporting API

**Version** 0.5.0

**Author** Gaurav Sood [aut, cre]

**Maintainer** Gaurav Sood <gsood07@gmail.com>

**Description** Provides access to YouTube Analytics API v2 for retrieving YouTube Analytics data including views, engagement metrics, demographics, and revenue data. Supports OAuth 2.0 authentication and channel group management. See <<https://developers.google.com/youtube/analytics/>> for API documentation.

**URL** <https://github.com/gojiplus/tubern>

**BugReports** <https://github.com/gojiplus/tubern/issues>

**Depends** R (>= 4.1.0)

**License** MIT + file LICENSE

**Encoding** UTF-8

**Language** en-US

**Imports** httr, jsonlite, checkmate, rlang (>= 1.1.0)

**VignetteBuilder** knitr

**Suggests** knitr (>= 1.11), testthat, httpptest, rmarkdown, xml2, lintr, covr, ggplot2, tibble

**RoxygenNote** 7.3.3

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2026-04-12 16:00:02 UTC

## Contents

add_groups . . . . .	2
add_group_item . . . . .	3
check_api_quota . . . . .	4
data_transformation . . . . .	4

date_helpers . . . . .	4
delete_group . . . . .	5
delete_group_item . . . . .	5
diagnose_tubern . . . . .	6
error_handling . . . . .	7
get_audience_demographics . . . . .	7
get_available_dimensions . . . . .	8
get_available_metrics . . . . .	9
get_channel_overview . . . . .	9
get_common_date_ranges . . . . .	10
get_daily_performance . . . . .	11
get_geographic_performance . . . . .	12
get_report . . . . .	13
get_revenue_report . . . . .	15
get_top_videos . . . . .	16
list_groups . . . . .	17
list_group_items . . . . .	17
report_helpers . . . . .	18
resolve_date_range . . . . .	18
tubern . . . . .	19
tubern_DELETE . . . . .	19
tubern_GET . . . . .	20
tubern_POST . . . . .	20
tubern_PUT . . . . .	21
update_group . . . . .	21
validation_helpers . . . . .	22
yt_check_token . . . . .	22
yt_export_csv . . . . .	22
yt_extract_summary . . . . .	23
yt_oauth . . . . .	24
yt_quick_plot . . . . .	25
yt_to_dataframe . . . . .	26
yt_to_tibble . . . . .	26

<b>Index</b>	<b>28</b>
--------------	-----------

---

add_groups	<i>Add Groups</i>
------------	-------------------

---

### Description

Creates a new channel group.

### Usage

```
add_groups(resource_details, ...)
```

**Arguments**

resource\_details  
Named nested list. Required. Must provide: snippet with title (required), and optionally description and contentDetails with itemType

... Additional arguments passed to [tubern\\_POST](#).

**Value**

named list

**References**

<https://developers.google.com/youtube/analytics/reference/groups/insert>

**Examples**

```
## Not run:
add_groups(list(
  snippet = list(
    title = "My Channel Group",
    description = "A group for organizing channels"
  ),
  contentDetails = list(itemType = "youtube#channel")
))

## End(Not run)
```

---

add_group_item	<i>Add Group Item</i>
----------------	-----------------------

---

**Description**

Add Group Item

**Usage**

```
add_group_item(resource_details, ...)
```

**Arguments**

resource\_details  
Named nested list. Required. Must provide: groupId, resource.id

... Additional arguments passed to [tubern\\_POST](#).

**Value**

named list

**References**

<https://developers.google.com/youtube/analytics/reference/groupItems/insert>

**Examples**

```
## Not run:
add_group_item(list(groupId = "", resource.id="hello"))

## End(Not run)
```

---

check\_api\_quota      *Check API quota status and provide guidance*

---

**Description**

Check API quota status and provide guidance

**Usage**

```
check_api_quota()
```

**Value**

Message about current quota usage (if available)

**Examples**

```
## Not run:
check_api_quota()

## End(Not run)
```

---

data\_transformation      *Data Transformation Utilities*

---

**Description**

Functions to transform YouTube Analytics API responses into common R data formats

---

date\_helpers      *Date Helper Functions for tubern*

---

**Description**

Convenience functions for handling dates in YouTube Analytics API requests

---

delete_group	<i>Delete Group</i>
--------------	---------------------

---

**Description**

Delete Group

**Usage**

```
delete_group(id, ...)
```

**Arguments**

id	String. Required. “The id parameter specifies the YouTube group ID of the group that is being deleted.”
...	Additional arguments passed to <a href="#">tubern_DELETE</a> .

**Value**

named list

**References**

<https://developers.google.com/youtube/analytics/reference/groups/delete>

**Examples**

```
## Not run:  
delete_group(id="ABZZzGSIAAA")  
  
## End(Not run)
```

---

delete_group_item	<i>Delete Group Item</i>
-------------------	--------------------------

---

**Description**

Delete Group Item

**Usage**

```
delete_group_item(id, ...)
```

**Arguments**

`id` String. Required. “The id parameter specifies the YouTube group item ID for the group that is being deleted.”

`...` Additional arguments passed to `tubern_DELETE`.

**Value**

named list

**References**

<https://developers.google.com/youtube/analytics/reference/groupItems/delete>

**Examples**

```
## Not run:  
delete_group_item(id="ABZZzGSIAAA")  
  
## End(Not run)
```

---

diagnose_tubern	<i>Diagnose common tubern issues</i>
-----------------	--------------------------------------

---

**Description**

Diagnose common tubern issues

**Usage**

```
diagnose_tubern()
```

**Value**

Diagnostic information about tubern setup

**Examples**

```
## Not run:  
diagnose_tubern()  
  
## End(Not run)
```

---

error_handling	<i>Enhanced Error Handling for tubern</i>
----------------	---

---

**Description**

Custom error classes and improved error handling for YouTube Analytics API using rlang for modern error handling patterns.

---

get_audience_demographics	<i>Get audience demographics report</i>
---------------------------	---

---

**Description**

Retrieves demographic breakdown of your audience by age and gender.

**Usage**

```
get_audience_demographics(
  date_range,
  end_date = NULL,
  ids = "channel==MINE",
  dimension = c("ageGroup", "gender"),
  metrics = c("views", "estimatedMinutesWatched"),
  ...
)
```

**Arguments**

date_range	Date range string or start date
end_date	End date (only needed if date_range is a specific start date)
ids	Channel identifier (default: "channel==MINE")
dimension	Demographic dimension: "ageGroup", "gender", or both (default: both)
metrics	Vector of metrics to include (default: views, estimatedMinutesWatched)
...	Additional arguments passed to get_report()

**Value**

API response with demographic data

**Examples**

```
## Not run:  
# Full demographic breakdown  
get_audience_demographics("last_90_days")  
  
# Age groups only  
get_audience_demographics("this_month", dimension = "ageGroup")  
  
# Gender breakdown only  
get_audience_demographics("this_quarter", dimension = "gender")  
  
## End(Not run)
```

---

get\_available\_dimensions

*Get available dimensions with descriptions*

---

**Description**

Get available dimensions with descriptions

**Usage**

```
get_available_dimensions(pattern = NULL)
```

**Arguments**

pattern            Optional regex pattern to filter dimensions

**Value**

Named character vector of dimensions and descriptions

**Examples**

```
# Get all dimensions  
get_available_dimensions()  
  
# Get only geographic dimensions  
get_available_dimensions("country|city")
```

---

get\_available\_metrics *Get available metrics with descriptions*

---

**Description**

Get available metrics with descriptions

**Usage**

```
get_available_metrics(pattern = NULL)
```

**Arguments**

pattern            Optional regex pattern to filter metrics

**Value**

Named character vector of metrics and descriptions

**Examples**

```
# Get all metrics
get_available_metrics()

# Get only view-related metrics
get_available_metrics("view")
```

---

get\_channel\_overview *Get channel overview report*

---

**Description**

Retrieves key channel performance metrics for a specified time period.

**Usage**

```
get_channel_overview(
  date_range,
  end_date = NULL,
  ids = "channel==MINE",
  include_engagement = TRUE,
  include_subscribers = TRUE,
  ...
)
```

**Arguments**

<code>date_range</code>	Date range string like "last_30_days" or start date for custom range
<code>end_date</code>	End date (only needed if <code>date_range</code> is a specific start date)
<code>ids</code>	Channel identifier (default: "channel==MINE")
<code>include_engagement</code>	Logical. Include likes, dislikes, comments, shares (default: TRUE)
<code>include_subscribers</code>	Logical. Include subscriber metrics (default: TRUE)
<code>...</code>	Additional arguments passed to <code>get_report()</code>

**Value**

API response with channel overview data

**Examples**

```
## Not run:
# Channel performance for last 30 days
get_channel_overview("last_30_days")

# Channel performance for specific date range
get_channel_overview("2024-01-01", end_date = "2024-01-31")

# Basic metrics only
get_channel_overview("this_month", include_engagement = FALSE, include_subscribers = FALSE)

## End(Not run)
```

---

```
get_common_date_ranges
```

*Get common date ranges*

---

**Description**

Get common date ranges

**Usage**

```
get_common_date_ranges()
```

**Value**

Named list of common date ranges

**Examples**

```
get_common_date_ranges()
```

---

get\_daily\_performance *Get daily performance time series*

---

## Description

Retrieves day-by-day performance metrics for trend analysis.

## Usage

```
get_daily_performance(  
  date_range,  
  end_date = NULL,  
  ids = "channel==MINE",  
  metrics = c("views", "estimatedMinutesWatched"),  
  ...  
)
```

## Arguments

date_range	Date range string or start date
end_date	End date (only needed if date_range is a specific start date)
ids	Channel identifier (default: "channel==MINE")
metrics	Vector of metrics to include (default: views, estimatedMinutesWatched)
...	Additional arguments passed to get_report()

## Value

API response with daily time series data

## Examples

```
## Not run:  
# Daily views for last 30 days  
get_daily_performance("last_30_days")  
  
# Daily performance with engagement metrics  
get_daily_performance("this_month", metrics = c("views", "likes", "comments", "shares"))  
  
## End(Not run)
```

---

`get_geographic_performance`*Get geographic performance report*

---

### Description

Retrieves performance metrics broken down by country or other geographic dimensions.

### Usage

```
get_geographic_performance(  
  date_range,  
  end_date = NULL,  
  ids = "channel==MINE",  
  dimension = "country",  
  metrics = c("views", "estimatedMinutesWatched"),  
  max_results = 25,  
  ...  
)
```

### Arguments

<code>date_range</code>	Date range string or start date
<code>end_date</code>	End date (only needed if <code>date_range</code> is a specific start date)
<code>ids</code>	Channel identifier (default: "channel==MINE")
<code>dimension</code>	Geographic dimension: "country", "province", or "city" (default: "country")
<code>metrics</code>	Vector of metrics to include (default: views, estimatedMinutesWatched)
<code>max_results</code>	Number of results to return (default: 25)
<code>...</code>	Additional arguments passed to <code>get_report()</code>

### Value

API response with geographic data

### Examples

```
## Not run:  
# Top countries by views  
get_geographic_performance("last_30_days")  
  
# US states/provinces (requires US-only data)  
get_geographic_performance("this_month", dimension = "province",  
  filters = "country==US")  
  
## End(Not run)
```

get\_report

*Get Reports***Description**

Retrieves YouTube Analytics reports containing YouTube Analytics data.

**Usage**

```
get_report(
  ids,
  metrics,
  start_date = NULL,
  end_date = NULL,
  currency = NULL,
  dimensions = NULL,
  filters = NULL,
  include_historical_channel_data = NULL,
  max_results = NULL,
  sort = NULL,
  start_index = NULL,
  ...
)
```

**Arguments**

ids	String. Channel or content owner identifier. For channels: "channel==MINE" (for authenticated user's channel) or "channel==UCxxxxxxxxxxxxxxxx" (for specific channel ID). For content owners: "contentOwner==ownerName".
metrics	String. Comma-separated list of YouTube Analytics metrics, such as views or likes, dislikes.
start_date	String. Start date for the report. Can be in YYYY-MM-DD format or relative date like "last_30_days", "this_month", "yesterday".
end_date	String. End date for the report. Can be in YYYY-MM-DD format or relative date. If NULL and start_date is relative, will be calculated automatically.
currency	Optional. String. Default is USD. Specifies what earnings metrics like earnings, adEarnings, grossRevenue, playbackBasedCpm, impressionBasedCpm will be reported in.
dimensions	String. Optional. Comma-separated list of YouTube Analytics dimensions, such as video or ageGroup, gender.
filters	String. Optional. Dimension value filters. Multiple filters can be separated by semicolons. For video filtering: "video==videoId1,videoId2" (comma-separated video IDs). For country filtering: "country==US". Combined example: "video==videoId1,videoId2;country==US". Note: When filtering by video IDs, ensure the dimensions parameter includes "video".

include_historical_channel_data	Boolean. Default is FALSE. “Whether the API response should include channels’ watch time and view data from the time period prior to when the channels were linked to the content owner.”
max_results	Integer. Optional. The maximum number of rows to include in the response.
sort	String. Optional. A comma-separated list of dimensions or metrics that determine the sort order for YouTube
start_index	Integer. Optional. “The 1-based index of the first entity to retrieve.”
...	Additional arguments passed to <a href="#">tubern_GET</a> .

**Value**

named list

**Troubleshooting 404 Errors**

If you encounter a 404 "Not Found" error, check the following:

- Ensure YouTube Analytics API is enabled in your Google Cloud Console project
- Verify your authentication token is valid and has the correct scopes
- Check that the channel ID (if using specific channel ID) exists and you have access to it
- Use "channel==MINE" to access your own authenticated channel’s data

**References**

<https://developers.google.com/youtube/analytics/reference/reports/query>

**Examples**

```
## Not run:
# Basic channel report
get_report(ids = "channel==MINE", metrics = "views",
           start_date = "2020-01-01", end_date = "2020-01-31")

# Report with video filtering (requires dimensions = "video")
get_report(ids = "channel==MINE",
           metrics = "views,likes,comments",
           dimensions = "video",
           filters = "video==videoId1,videoId2",
           start_date = "2020-01-01", end_date = "2020-01-31")

# Report with country filtering
get_report(ids = "channel==MINE",
           metrics = "views",
           filters = "country==US",
           start_date = "2020-01-01", end_date = "2020-01-31")

## End(Not run)
```

---

get\_revenue\_report      *Get revenue report (requires monetary scope)*

---

## Description

Retrieves revenue and monetization metrics.

## Usage

```
get_revenue_report(  
  date_range,  
  end_date = NULL,  
  ids = "channel==MINE",  
  currency = "USD",  
  include_cpm = TRUE,  
  ...  
)
```

## Arguments

date_range	Date range string or start date
end_date	End date (only needed if date_range is a specific start date)
ids	Channel identifier (default: "channel==MINE")
currency	Currency code (default: "USD")
include_cpm	Logical. Include CPM metrics (default: TRUE)
...	Additional arguments passed to get_report()

## Value

API response with revenue data

## Examples

```
## Not run:  
# Revenue report for last month  
get_revenue_report("last_month")  
  
# Revenue in different currency  
get_revenue_report("this_quarter", currency = "EUR")  
  
## End(Not run)
```

---

get_top_videos	<i>Get top videos report</i>
----------------	------------------------------

---

### Description

Retrieves performance metrics for individual videos, sorted by views.

### Usage

```
get_top_videos(
  date_range,
  end_date = NULL,
  ids = "channel==MINE",
  max_results = 10,
  metrics = c("views", "likes", "comments"),
  ...
)
```

### Arguments

date_range	Date range string or start date
end_date	End date (only needed if date_range is a specific start date)
ids	Channel identifier (default: "channel==MINE")
max_results	Number of top videos to return (default: 10)
metrics	Vector of metrics to include (default: views, likes, comments)
...	Additional arguments passed to get_report()

### Value

API response with top videos data

### Examples

```
## Not run:
# Top 10 videos by views in last 30 days
get_top_videos("last_30_days")

# Top 25 videos with more metrics
get_top_videos("this_month", max_results = 25,
               metrics = c("views", "likes", "comments", "shares", "estimatedMinutesWatched"))

## End(Not run)
```

---

list_groups	<i>List Groups</i>
-------------	--------------------

---

**Description**

List Groups

**Usage**

```
list_groups(filter, page_token = NULL, ...)
```

**Arguments**

filter	Named Vector. Required. Only one of the two: id or mine. id: Comma-separated list of YouTube group ID(s) to retrieve. mine: Set to TRUE to retrieve all groups owned by the authenticated user.
page_token	String. Optional. Identifies a specific page in the result set that should be returned.
...	Additional arguments passed to <a href="#">tubern_GET</a> .

**Value**

named list

**References**

<https://developers.google.com/youtube/analytics/reference/groups/list>

**Examples**

```
## Not run:  
list_groups(filter = c(mine = TRUE))  
  
## End(Not run)
```

---

list_group_items	<i>List Group Items</i>
------------------	-------------------------

---

**Description**

List Group Items

**Usage**

```
list_group_items(group_id, ...)
```

**Arguments**

group\_id      String. Required. ID of the group  
 ...            Additional arguments passed to `tubern_GET`.

**Value**

named list

**References**

<https://developers.google.com/youtube/analytics/reference/groupItems/list>

**Examples**

```
## Not run:
list_group_items(group_id = "vponEBg8hrR1yBUX0Hz66Uc5WMk/vyGp6PvFo4RvsFtPoIWeCReyIC8")

## End(Not run)
```

---

report\_helpers      *Common Report Helper Functions*

---

**Description**

Pre-configured functions for common YouTube Analytics report types

---

resolve\_date\_range      *Resolve date range with support for relative dates*

---

**Description**

Resolve date range with support for relative dates

**Usage**

```
resolve_date_range(start_date, end_date = NULL)
```

**Arguments**

start\_date      Start date (can be relative like "last\_30\_days" or absolute like "2024-01-01")  
 end\_date        End date (can be relative or absolute). If NULL and start\_date is relative, will  
 be calculated automatically

**Value**

List with resolved start\_date and end\_date as YYYY-MM-DD strings

**Examples**

```
# Relative date ranges
resolve_date_range("last_30_days")
resolve_date_range("this_month")
resolve_date_range("last_quarter")

# Absolute date ranges
resolve_date_range("2024-01-01", "2024-01-31")
```

tubern

**tubern** provides access to the YouTube Analytics and Reporting API**Description**

Provides access to YouTube Analytics API v2 for retrieving YouTube Analytics data including views, engagement metrics, demographics, and revenue data. Supports OAuth 2.0 authentication and channel group management. See <https://developers.google.com/youtube/analytics/> for API documentation.

**Author(s)**

**Maintainer:** Gaurav Sood <gsood07@gmail.com>

**See Also**

Useful links:

- <https://github.com/gojplus/tubern>
- Report bugs at <https://github.com/gojplus/tubern/issues>

tubern\_DELETE

*DELETE***Description**

DELETE

**Usage**

```
tubern_DELETE(path, query = NULL, body = "", ...)
```

**Arguments**

path	path to specific API request URL
query	query list
body	passing image through body
...	Additional arguments passed to <a href="#">DELETE</a> .

**Value**

list

---

`tubern_GET`*Base POST AND GET functions. Not exported.*

---

**Description**

GET

**Usage**`tubern_GET(path, query = NULL, ...)`**Arguments**

<code>path</code>	path to specific API request URL
<code>query</code>	query list
<code>...</code>	Additional arguments passed to <a href="#">GET</a> .

**Value**

list

---

`tubern_POST`*POST*

---

**Description**

POST

**Usage**`tubern_POST(path, query = NULL, body = "", ...)`**Arguments**

<code>path</code>	path to specific API request URL
<code>query</code>	query list
<code>body</code>	passing image through body
<code>...</code>	Additional arguments passed to <a href="#">POST</a> .

**Value**

list

---

tubern_PUT	<i>PUT</i>
------------	------------

---

**Description**

PUT

**Usage**

```
tubern_PUT(path, query = NULL, body = "", ...)
```

**Arguments**

path	path to specific API request URL
query	query list
body	passing image through body
...	Additional arguments passed to <a href="#">PUT</a> .

**Value**

list

---

update_group	<i>Update Groups</i>
--------------	----------------------

---

**Description**

Update Groups

**Usage**

```
update_group(resource_details, ...)
```

**Arguments**

resource_details	Named nested list. Required. Must provide: id, snippet title
...	Additional arguments passed to <a href="#">tubern_PUT</a> .

**Value**

named list

**References**

<https://developers.google.com/youtube/analytics/reference/groups/update>

**Examples**

```
## Not run:
update_group(list(id="ABZZzGSIAAA", snippet = list(title = "hello")))

## End(Not run)
```

---

validation_helpers	<i>YouTube Analytics API Validation Helpers</i>
--------------------	---

---

**Description**

Internal validation functions for metrics, dimensions, and parameters using checkmate for robust parameter validation.

---

yt_check_token	<i>Check if authentication token is in options</i>
----------------	--

---

**Description**

Check if authentication token is in options

**Usage**

```
yt_check_token()
```

---

yt_export_csv	<i>Export data to CSV</i>
---------------	---------------------------

---

**Description**

Export data to CSV

**Usage**

```
yt_export_csv(api_response, filename = NULL, ...)
```

**Arguments**

api_response	API response from YouTube Analytics
filename	Output filename (default: auto-generated based on timestamp)
...	Additional arguments passed to yt_to_dataframe()

**Value**

Path to saved file

**Examples**

```
## Not run:  
report <- get_daily_performance("last_30_days")  
file_path <- yt_export_csv(report, "daily_performance.csv")  
  
## End(Not run)
```

---

yt\_extract\_summary     *Extract summary statistics from API response*

---

**Description**

Extract summary statistics from API response

**Usage**

```
yt_extract_summary(api_response)
```

**Arguments**

api\_response     API response from YouTube Analytics

**Value**

Named list with summary statistics

**Examples**

```
## Not run:  
report <- get_channel_overview("last_30_days")  
summary <- yt_extract_summary(report)  
print(summary)  
  
## End(Not run)
```

yt\_oauth

*Set up Authorization***Description**

Simplified OAuth2 setup for YouTube Analytics API. This function will automatically detect the required scope based on your needs and provide helpful setup guidance.

**Usage**

```
yt_oauth(
  app_id = NULL,
  app_secret = NULL,
  scope = "analytics",
  token = ".httr-oauth",
  setup_guide = interactive(),
  ...
)
```

**Arguments**

app_id	Client ID from Google Cloud Console; required; no default
app_secret	Client secret from Google Cloud Console; required; no default
scope	Character. One of: <ul style="list-style-type: none"> <li>"analytics" - Basic analytics data (views, likes, etc.) - Default</li> <li>"monetary" - Includes revenue and monetization data</li> <li>"auto" - Automatically detect scope based on first API call</li> </ul>
token	Path to file containing the token. Default is .httr-oauth in working directory.
setup_guide	Logical. Show setup guide for first-time users (default: TRUE for interactive sessions)
...	Additional arguments passed to <code>oauth2.0_token</code>

**Details**

The function looks for .httr-oauth in the working directory. If it doesn't find it, it expects an application ID and a secret. The function launches a browser to allow you to authorize the application.

**Value**

Sets the google\_token option and saves .httr-oauth in working directory

**References**

<https://developers.google.com/youtube/analytics/reference/>

## Examples

```
## Not run:
# Basic setup (will show setup guide first time)
yt_oauth("your-client-id.apps.googleusercontent.com", "your-client-secret")

# Setup with monetary scope for revenue data
yt_oauth("your-client-id.apps.googleusercontent.com", "your-client-secret",
         scope = "monetary")

# Skip setup guide
yt_oauth("your-client-id.apps.googleusercontent.com", "your-client-secret",
         setup_guide = FALSE)

## End(Not run)
```

---

yt_quick_plot	<i>Create a quick visualization of the data (if ggplot2 is available)</i>
---------------	---

---

## Description

Create a quick visualization of the data (if ggplot2 is available)

## Usage

```
yt_quick_plot(api_response, x_col = NULL, y_col = NULL, chart_type = "auto")
```

## Arguments

api_response	API response from YouTube Analytics
x_col	Column name for x-axis (auto-detected if NULL)
y_col	Column name for y-axis (auto-detected if NULL)
chart_type	Type of chart: "line", "bar", "point" (default: auto)

## Value

ggplot object or base R plot if ggplot2 not available

## Examples

```
## Not run:
# Daily views over time
daily_report <- get_daily_performance("last_30_days")
yt_quick_plot(daily_report)

# Top videos by views
top_videos <- get_top_videos("last_7_days")
yt_quick_plot(top_videos, chart_type = "bar")

## End(Not run)
```

---

yt\_to\_dataframe      *Convert API response to data frame*

---

### Description

Transforms YouTube Analytics API response into a clean data.frame with proper column names and types.

### Usage

```
yt_to_dataframe(api_response, clean_names = TRUE, parse_dates = TRUE)
```

### Arguments

api\_response      List returned from get\_report() or other API functions  
clean\_names       Logical. Clean column names by removing special characters (default: TRUE)  
parse\_dates       Logical. Parse date columns to Date objects (default: TRUE)

### Value

data.frame with transformed data, or NULL if no data available

### Examples

```
## Not run:  
# Get data and convert to data.frame  
report <- get_channel_overview("last_30_days")  
df <- yt_to_dataframe(report)  
  
# Keep original column names  
df <- yt_to_dataframe(report, clean_names = FALSE)  
  
## End(Not run)
```

---

yt\_to\_tibble          *Convert API response to tibble (if tibble is available)*

---

### Description

Convert API response to tibble (if tibble is available)

### Usage

```
yt_to_tibble(api_response, ...)
```

**Arguments**

`api_response`    API response from YouTube Analytics  
...                Additional arguments passed to `yt_to_dataframe()`

**Value**

tibble or `data.frame` if tibble not available

**Examples**

```
## Not run:  
report <- get_top_videos("last_7_days")  
tbl <- yt_to_tibble(report)  
  
## End(Not run)
```

# Index

add\_group\_item, [3](#)  
add\_groups, [2](#)

check\_api\_quota, [4](#)

data\_transformation, [4](#)  
date\_helpers, [4](#)  
DELETE, [19](#)  
delete\_group, [5](#)  
delete\_group\_item, [5](#)  
diagnose\_tubern, [6](#)

error\_handling, [7](#)

GET, [20](#)  
get\_audience\_demographics, [7](#)  
get\_available\_dimensions, [8](#)  
get\_available\_metrics, [9](#)  
get\_channel\_overview, [9](#)  
get\_common\_date\_ranges, [10](#)  
get\_daily\_performance, [11](#)  
get\_geographic\_performance, [12](#)  
get\_report, [13](#)  
get\_revenue\_report, [15](#)  
get\_top\_videos, [16](#)

list\_group\_items, [17](#)  
list\_groups, [17](#)

oauth2.0\_token, [24](#)

POST, [20](#)  
PUT, [21](#)

report\_helpers, [18](#)  
resolve\_date\_range, [18](#)

tubern, [19](#)  
tubern-package (tubern), [19](#)  
tubern\_DELETE, [5](#), [6](#), [19](#)  
tubern\_GET, [14](#), [17](#), [18](#), [20](#)  
tubern\_POST, [3](#), [20](#)  
tubern\_PUT, [21](#), [21](#)

update\_group, [21](#)

validation\_helpers, [22](#)

yt\_check\_token, [22](#)  
yt\_export\_csv, [22](#)  
yt\_extract\_summary, [23](#)  
yt\_oauth, [24](#)  
yt\_quick\_plot, [25](#)  
yt\_to\_dataframe, [26](#)  
yt\_to\_tibble, [26](#)