

# Package ‘rtypeform’

May 9, 2026

**Type** Package

**Title** Interface to 'typeform' Results

**Version** 2.1.1

**Maintainer** Colin Gillespie <csgillespie@gmail.com>

**Description** An R interface to the 'typeform' <<https://www.typeform.com/>> application program interface. Also provides functions for downloading your results.

**License** GPL-2 | GPL-3

**URL** <https://github.com/csgillespie/rtypeform>,  
<http://csgillespie.github.io/rtypeform/>

**BugReports** <https://github.com/csgillespie/rtypeform/issues>

**Imports** dplyr (>= 1.0.0), glue, httr, jsonlite, lubridate, purrr,  
rlang, tidyr

**Suggests** testthat

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**NeedsCompilation** no

**Author** Colin Gillespie [aut, cre],  
Jamie Owen [aut]

**Repository** CRAN

**Date/Publication** 2025-01-29 09:30:02 UTC

## Contents

check_api_response . . . . .	2
create_theme . . . . .	2
delete_form . . . . .	4
delete_responses . . . . .	4
get_api . . . . .	5
get_authorization . . . . .	5

get_me . . . . .	6
get_number_of_forms . . . . .	6
get_number_of_seats . . . . .	7
get_number_of_workspace . . . . .	7
get_response . . . . .	8
get_responses . . . . .	9
make_new_token . . . . .	10
post_response . . . . .	10
rtypeform_set_scope . . . . .	11

## Index 12

---

check_api_response	<i>Check api response</i>
--------------------	---------------------------

---

### Description

A function that checks the api response

### Usage

```
check_api_response(resp, content)
```

### Arguments

resp	A response
content	The content

---

create_theme	<i>Theme functions</i>
--------------	------------------------

---

### Description

Theme API functions

### Usage

```
create_theme(
  name,
  api = NULL,
  background_href = NULL,
  background_layout = "fullscreen",
  background_brightness = 0,
  colors_answer = "#4FB0AE",
  colors_background = "#FFFFFF",
  colors_button = "#4FB0AE",
```

```

    colors_question = "#3D3D3D",
    font = "Source Sans Pro",
    has_transparent_button = TRUE
)

get_theme(theme_id, api = NULL)

get_themes(api = NULL, page = 1, page_size = 10)

update_theme(
  theme_id,
  name,
  api = NULL,
  background_href = NULL,
  background_layout = "fullscreen",
  background_brightness = 0,
  colors_answer = "#4FB0AE",
  colors_background = "#FFFFFF",
  colors_button = "#4FB0AE",
  colors_question = "#3D3D3D",
  font = "Source Sans Pro",
  has_transparent_button = TRUE
)

delete_theme(theme_id, api = NULL)

```

### Arguments

name	See <a href="https://www.typeform.com/developers/create/reference/create-theme/">https://www.typeform.com/developers/create/reference/create-theme/</a>
api	Default NULL. Your private api key. If api is NULL, the environment variable <code>Sys.getenv("typeform_api2")</code> is used.
background_href	See <a href="https://www.typeform.com/developers/create/reference/create-theme/">https://www.typeform.com/developers/create/reference/create-theme/</a>
background_layout	See <a href="https://www.typeform.com/developers/create/reference/create-theme/">https://www.typeform.com/developers/create/reference/create-theme/</a>
background_brightness	See <a href="https://www.typeform.com/developers/create/reference/create-theme/">https://www.typeform.com/developers/create/reference/create-theme/</a>
colors_answer	See <a href="https://www.typeform.com/developers/create/reference/create-theme/">https://www.typeform.com/developers/create/reference/create-theme/</a>
colors_background	See <a href="https://www.typeform.com/developers/create/reference/create-theme/">https://www.typeform.com/developers/create/reference/create-theme/</a>
colors_button	See <a href="https://www.typeform.com/developers/create/reference/create-theme/">https://www.typeform.com/developers/create/reference/create-theme/</a>
colors_question	See <a href="https://www.typeform.com/developers/create/reference/create-theme/">https://www.typeform.com/developers/create/reference/create-theme/</a>
font	See <a href="https://www.typeform.com/developers/create/reference/create-theme/">https://www.typeform.com/developers/create/reference/create-theme/</a>
has_transparent_button	See <a href="https://www.typeform.com/developers/create/reference/create-theme/">https://www.typeform.com/developers/create/reference/create-theme/</a>

theme_id	The theme id
page	The page of results to retrieve. Default 1 is the first page of results.
page_size	Number of results to retrieve per page. Default is 10. Maximum is 200.

---

delete\_form                      *Deletes a typeform*

---

### Description

Delete the specified typeform.

### Usage

```
delete_form(form_id, api = NULL)
```

### Arguments

form_id	The form id of the typeform you want the results for.
api	Default NULL. Your private api key. If api is NULL, the environment variable <code>Sys.getenv("typeform_api2")</code> is used.

---

delete\_responses                *Delete Responses*

---

### Description

Delete entries in a type form- use with care!

### Usage

```
delete_responses(form_id, api = NULL, included_tokens = NULL)
```

### Arguments

form_id	The form id of the typeform you want the results for.
api	Default NULL. Your private api key. If api is NULL, the environment variable <code>Sys.getenv("typeform_api2")</code> is used.
included_tokens	A vector of tokens to delete. Maximum 1000. Token ids can be extracted from the meta data frame via <code>get_form</code> .

### Value

If successful, the function will return (invisibly) TRUE

---

get_api	<i>Retrieve API key from Renviron</i>
---------	---------------------------------------

---

**Description**

If the entry typeform\_api2 exists in your .Renviron return that value. Otherwise, raise an error.

**Usage**

```
get_api(api = NULL)
```

**Arguments**

api	Default NULL. Your private api key. If api is NULL, the environment variable Sys.getenv("typeform_api2") is used.
-----	---

**Details**

In version 1 of the typeform API, rtypeform looked for the key typeform\_api,

---

get_authorization	<i>Get Authorization</i>
-------------------	--------------------------

---

**Description**

A function to get authorization

**Usage**

```
get_authorization(api)
```

**Arguments**

api	An authentication key
-----	-----------------------

---

get_me	<i>Retrieve your own account information.</i>
--------	---

---

**Description**

Returns alias, email and language.

**Usage**

```
get_me(api = NULL)
```

**Arguments**

api	Default NULL. Your private api key. If api is NULL, the environment variable <code>Sys.getenv("typeform_api2")</code> is used.
-----	--

---

get_number_of_forms	<i>Fetch all available typeforms</i>
---------------------	--------------------------------------

---

**Description**

This function returns a two column data frame containing the typeform names and their associated ids.

**Usage**

```
get_number_of_forms(api = NULL, search = "", workspace_id = NULL)
```

```
get_forms(
  api = NULL,
  page = 1,
  page_size = 10,
  search = "",
  workspace_id = NULL
)
```

**Arguments**

api	Default NULL. Your private api key. If api is NULL, the environment variable <code>Sys.getenv("typeform_api")</code> is used.
search	Returns items that contain the specified string.
workspace_id	Retrieve typeforms for the specified workspace.
page	The page of results to retrieve. Default 1 is the first page of results.
page_size	Number of results to retrieve per page. Default is 10. Maximum is 200.

**Value**

A list containing content and the response.

---

get\_number\_of\_seats     *Get team members*

---

**Description**

Return a tibble containing the current team members. The total number of seats is attached via an attribute.

**Usage**

```
get_number_of_seats(api = NULL)
```

```
get_teams(api = NULL)
```

**Arguments**

api                     Default NULL. Your private api key. If api is NULL, the environment variable Sys.getenv("typeform\_api2") is used.

---

get\_number\_of\_workspace     *Available workspaces*

---

**Description**

A tibble containing details on workspaces

**Usage**

```
get_number_of_workspace(api = NULL, search = NULL)
```

```
get_workspaces(api = NULL, search = NULL, page = 1, page_size = 10)
```

```
create_workspace(workspace_name, api = NULL)
```

```
get_workspace(workspace_id, api = NULL)
```

```
update_workspace(  
  workspace_id,  
  api = NULL,  
  workspace_name = NULL,  
  add_members = NULL,
```

```

        remove_members = NULL
    )

delete_workspace(workspace_id, api = NULL)

```

### Arguments

api	Default NULL. Your private api key. If api is NULL, the environment variable Sys.getenv("typeform_api2") is used.
search	Returns items that contain the specified string.
page	The page of results to retrieve. Default 1 is the first page of results.
page_size	Number of results to retrieve per page. Default is 10. Maximum is 200.
workspace_name	The name workspace name
workspace_id	Retrieve typeforms for the specified workspace.
add_members	Email address
remove_members	Email address

---

get_response	<i>Get response</i>
--------------	---------------------

---

### Description

A function that allows you to get a response from the server.

### Usage

```
get_response(api, url)
```

### Arguments

api	An authentication key
url	The URL of the site

---

get_responses	<i>Download questionnaire results</i>
---------------	---------------------------------------

---

### Description

Download results for a particular typeform questionnaire.

### Usage

```
get_responses(
  form_id,
  api = NULL,
  page_size = 25,
  since = NULL,
  until = NULL,
  after = NULL,
  before = NULL,
  completed = NULL,
  query = NULL,
  fields = NULL
)
```

### Arguments

form_id	The form id of the typeform you want the results for.
api	Default NULL. Your private api key. If api is NULL, the environment variable <code>Sys.getenv("typeform_api2")</code> is used.
page_size	Maximum number of responses. Default value is 25. Maximum value is 1000.
since	default NULL. Fetch only the results after a specific date and time. If NULL return all results. This should be a date time object. The timezone of the object will be converted to UTC.
until	default NULL. Similar to since.
after	default NULL. Fetch only the results after a specific date and time. If NULL return all results. If you use the after parameter, the responses will be sorted in the order that our system processed them (instead of the default order, submitted_at). This ensures that you can traverse the complete set of responses without repeating entries.
before	default NULL. Similar to after
completed	default NULL, return all results. Fetch only completed results (TRUE), or only not-completed results (=FALSE). If NULL return all results. Warning. It's not possible to determine completed/non-completed results.
query	Limit request to only responses that that include the specified term.
fields	Not implemented. Pull requests welcome

**Value**

A list. The first value is meta information. Subsequent elements are questions..

**See Also**

[typeform/retrieve-responses](#)

---

make_new_token	<i>Create an OAuth token</i>
----------------	------------------------------

---

**Description**

Create a token object using the client id and secret options set, together with the defined scopes.

**Usage**

```
make_new_token()
```

---

post_response	<i>Post response</i>
---------------	----------------------

---

**Description**

A function that allows you to post an update to the server

**Usage**

```
post_response(api, url, body = NULL, ...)
```

**Arguments**

api	An authentication key
url	The URL of the site
body	The information sent out to the server. e.g. the title
...	Other arguments

---

rtypeform\_set\_scope    *Set access scopes for OAuth access*

---

**Description**

Define access scopes for an application. See <https://www.typeform.com/developers/get-started/scopes/> for more information on what scopes are allowed and their permissions.

Set option for rtypeform client id, secret

**Usage**

```
rtypeform_set_scope(scopes = NULL)
```

```
rtypeform_set_client_id(id = NULL)
```

```
rtypeform_set_client_secret(secret = NULL)
```

```
rtypeform_set_token(token)
```

**Arguments**

scopes	A character vector of scopes
id	A client id from a typeform registered application
secret	A client secret from a typeform registered application
token	Either a string of a personal access token, or an OAuth token returned by <code>make_new_token</code> .

**Examples**

```
rtypeform_set_scope(scopes = c("forms:read"))
```

# Index

check\_api\_response, 2  
create\_theme, 2  
create\_workspace  
    (get\_number\_of\_workspace), 7  
  
delete\_form, 4  
delete\_responses, 4  
delete\_theme (create\_theme), 2  
delete\_workspace  
    (get\_number\_of\_workspace), 7  
  
get\_api, 5  
get\_authorization, 5  
get\_forms (get\_number\_of\_forms), 6  
get\_me, 6  
get\_number\_of\_forms, 6  
get\_number\_of\_seats, 7  
get\_number\_of\_workspace, 7  
get\_response, 8  
get\_responses, 9  
get\_teams (get\_number\_of\_seats), 7  
get\_theme (create\_theme), 2  
get\_themes (create\_theme), 2  
get\_workspace  
    (get\_number\_of\_workspace), 7  
get\_workspaces  
    (get\_number\_of\_workspace), 7  
  
make\_new\_token, 10  
  
post\_response, 10  
  
rtypeform\_set\_client\_id  
    (rtypeform\_set\_scope), 11  
rtypeform\_set\_client\_secret  
    (rtypeform\_set\_scope), 11  
rtypeform\_set\_scope, 11  
rtypeform\_set\_token  
    (rtypeform\_set\_scope), 11  
  
update\_theme (create\_theme), 2  
  
update\_workspace  
    (get\_number\_of\_workspace), 7