Package ‘zdeskR’

October 14, 2022

Title Connect to Your 'Zendesk' Data

Version 0.2.0

Description Facilitates making a connection to the 'Zendesk' API and executing various queries. You can use it to get ticket data and ticket metrics. The 'Zendesk' documentation is available at <https://developer.zendesk.com/rest_api /docs/support/introduction>. This package is not supported by 'Zendesk' (owner of the software).

URL `https://github.com/chrisumphlett/zdeskR`

BugReports `https://github.com/chrisumphlett/zdeskR/issues`

License CC0

Encoding UTF-8

Imports dplyr (>= 1.0.0), magrittr (>= 1.5), jsonlite (>= 1.6.1), purrr (>= 0.3.3), httr (>= 1.4.1), tidyr (>= 1.0.0), plyr (>= 1.8.6)

RoxygenNote 7.1.2

NeedsCompilation no

Author Chris Umphlett [aut, cre], Avinash Panigrahi [aut]

Maintainer Chris Umphlett <`christopher.umphlett@gmail.com`>

Repository CRAN

Date/Publication 2022-03-16 22:20:02 UTC

R topics documented:

`get_all_ticket_metrics` *Get Metrics on All Zendesk Tickets*

Description

This function takes your Email Id, authentication token, sub-domain and parse all the tickets and its corresponding metrics in a list. Since each iteration only returns 100 tickets at a time you must run the loop until the "next_page" parameter is equal to null.

Usage

```
get_all_ticket_metrics(email_id, token, subdomain)
```

Arguments

`email_id`	Zendesk Email Id (username).
`token`	Zendesk API token.
`subdomain`	Your organization's Zendesk sub-domain.

Details

Its not a good practice to write down these authentication parameters in your code. There are various methods and packages available that are more secure; this package doesn't require you to use any one in particular.

Value

Data Frame with metrics for all tickets

References

https://developer.zendesk.com/rest_api/docs/support/ticket_metrics

Examples

```
## Not run:
ticket_metrics <- get_all_ticket_metrics(email_id, token, subdomain)

## End(Not run)
```

`get_custom_fields` *Returns the system and all the custom fields defined by your organization's zendesk administrator*

Description

It takes your Email Id, authentication token, sub-domain as parameters and gets the system and all the custom fields available for a zendesk ticket.

Usage

```
get_custom_fields(email_id, token, subdomain)
```

Arguments

`email_id`	Zendesk Email Id (username).
`token`	Zendesk API token.
`subdomain`	Your organization's Zendesk sub-domain.

Details

It's not a good practice to write down these authentication parameters in your code. There are various methods and packages available that are more secure; this package doesn't require you to use any one in particular.

Value

A data frame containing all ticket fields

References

https://developer.zendesk.com/rest_api/docs/support/ticket_fields

Examples

```
## Not run:
fields <- get_custom_fields(email_id, token, subdomain)

## End(Not run)
```

get_tickets *Get Zendesk Tickets*

Description

This function takes your Email Id, authentication token, sub-domain and start time as parameters and gets all the tickets which have been updated on or after the start time parameter. By default each page returns 1000 unique tickets and an "after_cursor" value which stores a pointer to the next page. After getting the first page it uses the pointer to fetch the subsequent pages.

Usage

```
get_tickets(
  email_id,
  token,
  subdomain,
  start_time,
  end_time,
  remove_cols = NULL
)
```

Arguments

`email_id`	Zendesk Email Id (username).
`token`	Zendesk API token.
`subdomain`	Your organization's Zendesk sub-domain.
`start_time`	String with a date or datetime to get all tickets modified after that date.
`end_time`	String with a date or datetime to get all tickets modified before that date.
`remove_cols`	Vector of column names to remove from the results.

Details

The start time parameter should be in 'UTC' format as Zendesk uses the 'UTC' time zone when retrieving tickets after the start time. For example, the US Eastern Time Zone is currently four hours being UTC. If one wanted to get tickets starting on August 1 at 12 am, you would need to enter "2020-08-01 04:00:00". The user must do proper adjustment to accommodate the time zone difference, if desired. A date can be provided, it will retrieve results as of 12 am in the UTC time zone.

Start and end times can be entered with or without the time component. End time cannot be in the future, but should work for values up to one minute prior to the current time.

It's not a good practice to write down these authentication parameters in your code. There are various methods and packages available that are more secure; this package doesn't require you to use any one in particular.

The remove_cols parameter allows the removal of custom fields causing errors. Errors occurred when a field was sometimes blank and assigned a logical type and then appended to non-blank, non-logical inside of purrr::map_dfr. See issue #1 on GH.

Value

a Data Frame containing all tickets after the start time.

References

https://developer.zendesk.com/rest_api/docs/support/incremental_export#start_time

Examples

```
## Not run:
all_tickets <- get_tickets(email_id, token, subdomain,
  start_time = "2021-01-31 00:00:00", end_time = "2021-01-31 23:59:59"
)

## End(Not run)
```

---

get_users *Returns All Available Zendesk Users.*

Description

It takes your Email Id, authentication token, sub-domain and parse all the users in a list. It iterates through all the pages returning only 100 users per page until the "next_page" parameter becomes null indicating there are no more pages to fetch.

Usage

```
get_users(email_id, token, subdomain, start_page = 1)
```

Arguments

`email_id`	Zendesk Email Id (username).
`token`	Zendesk API token.
`subdomain`	Your organization's Zendesk sub-domain.
`start_page`	First page of results to return, default is 1.

Details

It's not a good practice to write down these authentication parameters in your code. There are various methods and packages available that are more secure; this package doesn't require you to use any one in particular.

The start_page parameter is useful if you have many users. Each page contains 100 users. Zendesk does not have an incremental method for pulling users by date but after you retrieve all of your users once, you can then increment your start page to something that will limit the number of users you are re-pulling each time.

If you are pulling partial lists of users be aware that you will not get updates on older users. You will only get recently created users, not modified/deleted users and their modified data nor updated last login dates.

Value

Data Frame with user details

References

https://developer.zendesk.com/rest_api/docs/support/users

Examples

```
## Not run:
users <- get_users(email_id, token, subdomain)

## End(Not run)
```

Index