Package ‘rgtmx’

November 11, 2021

Type Package
Title Manage GTmetrix Tests in R
Version 0.1.4
Maintainer Roman A. Abashin <roman@nougat.ai>
Description This is a library to access the current API of the web speed test service 'GTmetrix'. It provides a convenient wrapper to start tests, get reports, and access all kinds of meta data. For more information about using the API please visit <https://gtmetrix.com/api/docs/2.0/>.
License MIT + file LICENSE
URL https://github.com/RomanAbashin/rgtmx
BugReports https://github.com/RomanAbashin/rgtmx/issues
Depends R (>= 3.1)
Imports httr, jsonlite
Encoding UTF-8
Suggests rmarkdown, knitr, testthat (>= 3.0.0), spelling
Config/testthat/edition 3
RoxygenNote 7.1.2
VignetteBuilder knitr
Language en-US
NeedsCompilation no
Author Roman A. Abashin [cre, aut]
Repository CRAN
Date/Publication 2021-11-11 19:10:02 UTC

R topics documented:

   check_input ................................................................. 2
get_account_status .......................................................... 3
get_all_tests ............................................................... 3
check_input

get_browser_details .............................................. 4
get_location_details ............................................. 4
get_report .......................................................... 5
get_test ............................................................ 6
show_available_browsers .......................................... 6
show_available_locations ......................................... 7
start_test .......................................................... 8

Index 11

check_input  Check variable inputs

Description
Internal function to check input variables

Usage
check_input(
  input,
  input_type,
  input_validation = NULL,
  min_value = -Inf,
  max_value = Inf,
  max_length = 1L,
  variable_name = NULL,
  is_missing = NULL
)

Arguments
input .
input_type .
input_validation .
min_value .
max_value .
max_length .
variable_name .
is_missing .

Value
nothing
get_account_status

Get the status of your GTmetrix account

Description
Show available credits and other meta data for the supplied API key.

Usage
get_account_status(api_key)

Arguments
api_key An active GTmetrix API key. (string)

Value
A data.frame that contains meta data of a GTmetrix account.

Examples
## Not run: output_table <- get_account_status(
  api_key = "API_KEY"
)
## End(Not run)

get_all_tests

Get a table of tests, their report IDs and other meta data.

Usage
get_all_tests(api_key, page_size = 50, page_number = 1)

Arguments
api_key An active GTmetrix API key. (string)
page_size Page size (default 50, max 500)
page_number Page (default 1)

Value
A data.frame object that contains test IDs and their meta data.
get_location_details

Description
Get details for a specific locations ID.

Usage
get_location_details(location, api_key)

Arguments
- location: Location ID. (integer)
- api_key: An active GTmetrix API key. (string)

Examples

```r
## Not run: output_table <- get_location_details(location = 3, api_key = "API_KEY")
## End(Not run)
```

get_browser_details

Description
Show browser details

Get details for a specific browsers ID.

Usage
get_browser_details(browser, api_key)

Arguments
- browser: Browser ID. (integer)
- api_key: An active GTmetrix API key. (string)

Value
A data.frame object that contains available browsers and their meta data.

Examples

```r
## Not run: output_table <- get_browser_details(  
  browser_id = 3, api_key = "API_KEY"
)
## End(Not run)
```
get_report

Value

A data.frame object that contains available locations and their meta data.

Examples

```r
## Not run: output_table <- get_location_details(
   location_id = 3, api_key = "API_KEY"
)
## End(Not run)
```

get_report

Get status and meta data of a specific report

Description

Get status and meta data of a specific GTmetrix report.

Usage

```r
get_report(report_id, api_key)
```

Arguments

- `report_id`: ID of a GTmetrix report. (string)
- `api_key`: An active GTmetrix API key. (string)

Value

A data.frame object that contains a GTmetrix report and its meta data.

Examples

```r
## Not run: output_table <- get_report(
   test_id = "REPORT_ID",
   api_key = "API_KEY"
)
## End(Not run)
```
get_test  

*Get status and meta data of a specific test*

**Description**

Get the status and meta data of a specific GTmetrix test. Returns the associated report instead, if the report is already completed.

**Usage**

```r
get_test(test_id, api_key, wait_for_completion = TRUE)
```

**Arguments**

- `test_id`: ID of a GTmetrix test. (string)
- `api_key`: An active GTmetrix API key. (string)
- `wait_for_completion`: Whether the function should wait for the completion of the test. If TRUE (default), the report associated with the test ID will be requested in roughly 3 second intervals and returned, when successful. If FALSE, the meta data of the test will be returned. (TRUE, FALSE)

**Value**

A data.frame object that contains either the test meta data or the GTmetrix report (if it’s already completed)

**Examples**

```r
## Not run: output_table <- get_test(
  test_id = "TEST_ID",
  api_key = "API_KEY"
)
## End(Not run)
```

---

**show_available_browsers**

*Show available browsers*

**Description**

Show available browsers for the supplied API key.

**Usage**

```r
show_available_browsers(api_key)
```
show_available_locations

Arguments

api_key An active GTmetrix API key. (string)

Value

A data.frame object that contains available browsers and their meta data.

Examples

## Not run: output_table <- show_available_browsers(api_key = "API_KEY")

show_available_locations

Show available locations

Description

Show available locations for the supplied API key.

Usage

show_available_locations(api_key)

Arguments

api_key An active GTmetrix API key. (string)

Value

A data.frame object that contains available locations and their meta data.

Examples

## Not run: output_table <- show_available_locations(api_key = "API_KEY")
Description

`start_test` starts a GTmetrix test and returns either the test itself (incl. meta data) or the associated report.

Usage

```r
start_test(
  url, 
  api_key, 
  wait_for_completion = TRUE, 
  location = 1, 
  browser = 3, 
  report = "lighthouse", 
  retention = 1, 
  httpauth_username = NULL, 
  httpauth_password = NULL, 
  adblock = 0, 
  cookies = NULL, 
  video = 0, 
  stop_onload = 0, 
  throttle = NULL, 
  allow_url = NULL, 
  block_url = NULL, 
  dns = NULL, 
  simulate_device = NULL, 
  user_agent = NULL, 
  browser_width = NULL, 
  browser_height = NULL, 
  browser_dppx = NULL, 
  browser_rotate = NULL
)
```

Arguments

- **url**: The URL of the page to test. (string)
- **api_key**: An active GTmetrix API key (string)
- **wait_for_completion**: Whether the function should wait for the completion of the test. If TRUE (default), the report associated with the test ID will be requested in roughly 3 second intervals and returned, when successful. If FALSE, the meta data of the test will be returned. (TRUE, FALSE)
- **location**: Location ID. Default = "1"
browser: Location ID. Default = "3"
report: A string for the type of report. "lighthouse" (default) for 'Lighthouse', "legacy" for 'PageSpeed'/"YSlow", "lighthouse,legacy" for both, "none" for a metrics-only report.
retention: Choose how long (in months) the report will be retained and accessible. Valid values: 1 (default), 6, 12, 24.
httpauth_username: Username for the test page HTTP access authentication. (string)
httpauth_password: Password for the test page HTTP access authentication. (string)
adblock: Enable AdBlock. 0 (default) = no, 1 = yes.
cookies: Specify cookies to supply with test page requests.
video: Enable generation of video. 0 (default) = no, 1 = yes
stop_onload: Stop the test at 'window.onload' instead of after the page has fully loaded (i.e. 2 seconds of network inactivity). 0 (default) = no, 1 = yes
throttle: Throttle the connection. Speed measured in Kbps, latency in ms. Format: "up/down/latency"
allow_url: Only load resources that match one of the URLs on this list. This uses the same syntax as the web front end.
block_url: Prevent loading of resources that match one of the URLs on this list. This occurs after the Only Allow URLs are applied. This uses the same syntax as the web front end.
dns: Use a custom DNS host and IP to run the test with.
simulate_device: Simulate the display of your site on a variety of devices using a pre-selected combination of Screen Resolutions, User Agents, and Device Pixel Ratios. (Expected: Device ID)
user_agent: Use a custom User Agent string.
browser_width: Set the width of the viewport for the analysis. Also requires browser_height to be set.
browser_height: Set the height of the viewport for the analysis. Also requires browser_width to be set.
browser_dppx: Set the device pixel ratio for the analysis. Decimals are allowed.
browser_rotate: Swaps the width and height of the viewport for the analysis. simulate_device overrides this parameter with preset values.

Value
A data.frame object that contains either the test meta data or the GTmetrix report (if it’s already completed).
Examples

```r
## Not run: output_table <- start_test(
  url = "google.com",
  api_key = "API_KEY",
  wait_for_completion = TRUE
)
## End(Not run)
```
Index

check_input, 2
get_account_status, 3
get_all_tests, 3
get_browser_details, 4
get_location_details, 4
get_report, 5
get_test, 6
show_available_browsers, 6
show_available_locations, 7
start_test, 8