Package ‘RAhrefs’

July 28, 2019

Type Package
Title 'Ahrefs' API R Interface
Version 0.1.4
Description Enables downloading detailed reports from <https://ahrefs.com> about backlinks from pointing to website, provides authentication with an API key as well as ordering, grouping and filtering functionalities.
License MIT + file LICENCE
URL https://ahrefs.com/
BugReports https://github.com/Leszek-Sieminski/RAhrefs/issues
Depends R (>= 3.4.0)
Imports assertthat, httr, jsonlite, testthat
NeedsCompilation no
Encoding UTF-8
LazyData true
RoxygenNote 6.1.1
Author Leszek Siemiński [aut, cre],
Performance Media Polska sp. z o.o. [cph, fnd]
Maintainer Leszek Siemiński <leszek.sieminski@gmail.com>
Repository CRAN
Date/Publication 2019-07-28 08:40:02 UTC

R topics documented:

ahrefs_metrics ......................................................... 2
ahrefs_reports .......................................................... 2
rah_ahrefs_rank .......................................................... 3
rah_anchors ............................................................... 5
rah_anchors_refdomains ............................................... 8
rah_auth ................................................................. 11
rah_backlinks ........................................................... 11
### ahrefs_metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>rah_backlinks_new_lost</td>
<td>14</td>
</tr>
<tr>
<td>rah_backlinks_new_lost_counters</td>
<td>17</td>
</tr>
<tr>
<td>rah_backlinks_one_per_domain</td>
<td>20</td>
</tr>
<tr>
<td>rah_broken_backlinks</td>
<td>23</td>
</tr>
<tr>
<td>rah_broken_links</td>
<td>26</td>
</tr>
<tr>
<td>rah_condition</td>
<td>29</td>
</tr>
<tr>
<td>rah_condition_set</td>
<td>31</td>
</tr>
<tr>
<td>rah_domain_rating</td>
<td>32</td>
</tr>
<tr>
<td>rah_downloader</td>
<td>34</td>
</tr>
<tr>
<td>rah_linked_anchors</td>
<td>36</td>
</tr>
<tr>
<td>rah_linked_domains</td>
<td>39</td>
</tr>
<tr>
<td>rah_linked_domains_by_type</td>
<td>42</td>
</tr>
<tr>
<td>rah_metrics</td>
<td>45</td>
</tr>
<tr>
<td>rah_metrics_extended</td>
<td>47</td>
</tr>
<tr>
<td>rah_pages</td>
<td>50</td>
</tr>
<tr>
<td>rah_pages_extended</td>
<td>52</td>
</tr>
<tr>
<td>rah_pages_info</td>
<td>55</td>
</tr>
<tr>
<td>rah_refdomains</td>
<td>58</td>
</tr>
<tr>
<td>rah_refdomains_by_type</td>
<td>61</td>
</tr>
<tr>
<td>rah_refdomains_new_lost</td>
<td>64</td>
</tr>
<tr>
<td>rah_refdomains_new_lost_counters</td>
<td>66</td>
</tr>
<tr>
<td>rah_refips</td>
<td>68</td>
</tr>
<tr>
<td>rah_subscription_info</td>
<td>71</td>
</tr>
</tbody>
</table>

#### Description

Description of Ahrefs metrics to include in requests with "where" & "having" usage possibility.

#### Usage

```r
data(ahrefs_metrics)
```

#### Format

An object of class 'data.frame'

#### Source

Ahrefs API Documentation[^1]

#### Examples

```r
data(ahrefs_metrics)
View(ahrefs_metrics)
```

[^1]: [https://ahrefs.com/api/documentation](https://ahrefs.com/api/documentation)
ahrefs_reports

Description

Description of all available Ahrefs reports provided with related function names

Usage

data(ahrefs_reports)

Format

An object of class 'data.frame'

Source

Ahrefs API Documentation

Examples

data(ahrefs_reports)
View(ahrefs_reports)

rah_ahrefs_rank

Export the URLs and their rankings.

Description

Export the URLs and their rankings.

Usage

rah_ahrefs_rank(target, token = Sys.getenv("AHREFS_AUTH_TOKEN"),
mode = "domain", metrics = NULL, limit = 1000, order_by = NULL,
where = NULL, having = NULL)

https://ahrefs.com/api/documentation
Arguments

- **target**: character string. Aim of a request: a domain, a directory or a URL.
- **token**: character string. Authentication token. Should be available through environmental variables after authentication with function `rah_auth()`.
- **mode**: character string. Mode of operation: exact, domain, subdomains or prefix. See more in Details section.
- **metrics**: character vector of columns to select. See more in Details section.
- **limit**: integer. Number of results to return.
- **order_by**: character vector of columns to sort on. See more in Details section.
- **where**: character string - a condition created by `rah_condition_set()` function that generates proper "where" condition to satisfy. See more in Details section.
- **having**: character string - a condition created by `rah_condition_set()` function that generates proper "having" condition to satisfy. See more in Details section.

Details

1. **available metrics** - you can select which columns (metrics) you want to download and which one would be useful in filtering, BUT not all of them can always be used in "where" & "having" conditions:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Where</th>
<th>Having</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Target of the request.</td>
</tr>
<tr>
<td>ahrefs_rank</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>URL Rating of the target.</td>
</tr>
</tbody>
</table>

2. "mode" parameter can take 4 different values that will affect how the results will be grouped. Example of URL directory with folder:

   - **Example URL**: ahrefs.com/api/
   - **exact**: ahrefs.com/api/
   - **domain**: ahrefs.com/*
   - **subdomains**: *ahrefs.com/*
   - **prefix**: ahrefs.com/api/*

Example of URL directory with subdomain:

   - **Example URL**: apiv2.ahrefs.com
   - **exact**: apiv2.ahrefs.com/
   - **domain**: apiv2.ahrefs.com/*
   - **subdomains**: *apiv2.ahrefs.com/*
   - **prefix**: apiv2.ahrefs.com/*

3. "order_by" parameter is a character string that forces sorting of the results. Structure:

   - **Structure**: "column_name:asc|desc"
- **Single column example:** "first_seen:asc" - this sorts results by first_seen column in ascending order
- **Multi column example:** "last_seen:desc,first_seen:asc" - this sorts results by 1) last_seen column in descending order, and next by 2) first_seen column in ascending order

4. "where" & "having" are **EXPERIMENTAL** parameters of condition sets (character strings) that control filtering the results. To create arguments:

1. use `rah_condition()` function to create a single condition, for example: `cond_1 <- rah_condition(column_name = "links", operator = "GREATER_THAN", value = "10")`
2. use `rah_condition_set()` function to group single conditions into final condition string, for example: `fin_cond <- rah_condition_set(cond_1, cond_2)`
3. provide final condition to proper report function as a parameter, for example: `RAhrefs::rah_ahrefs_rank(target = "ahrefs.com", token = "0123456789", mode = "domain", metrics = NULL, limit = 1000, where = fin_cond, order_by = "first_seen:asc")`

**Value**

data frame

**Source**

https://ahrefs.com/api/documentation

**See Also**

Other Ahrefs reports: `rah_anchors_refdomains`, `rah_anchors`, `rah_backlinks_new_lost_counters`, `rah_backlinks_new_lost`, `rah_backlinks_one_per_domain`, `rah_backlinks`, `rah_broken_backlinks`, `rah_broken_links`, `rah_domain_rating`, `rah_linked_anchors`, `rah_linked_domains_by_type`, `rah_linked_domains`, `rah_metrics_extended`, `rah_metrics`, `rah_pages_extended`, `rah_pages_info`, `rah_pages`, `rah_refdomains_by_type`, `rah_refdomains_new_lost_counters`, `rah_refdomains_new_lost`, `rah_refdomains`, `rah_refips`, `rah_subscription_info

**Examples**

```r
# Not run:
# creating single conditions for 'where' parameter
cond_1 <- RAhrefs::rah_condition(
  column_name = "url",
  operator = "WORD",
  value = "www")

cond_2 <- RAhrefs::rah_condition(
  column_name = "url",
  operator = "GREATER_THAN",
  value = "/cart")

# creating single conditions for 'having' parameter
cond_3 <- RAhrefs::rah_condition(
```
column_name = "ahrefs_rank",

# joining conditions into one 'where' condition set
cond_where <- RAhrefs::rah_condition_set(cond_1, cond_2)

# joining conditions into one 'having' condition set
cond_having <- RAhrefs::rah_condition_set(cond_3)

# downloading
b <- RAhrefs::rah_ahrefs_rank(
  target = "ahrefs.com",
  limit = 2,
  where = cond_where,
  having = cond_having,
  order_by = "ahrefs_rank:desc")

## End(Not run)

---

**rah_anchors**

Export the anchor text and the number of backlinks, referring pages and referring domains that has it.

**Description**

Export the anchor text and the number of backlinks, referring pages and referring domains that has it.

**Usage**

`rah_anchors(target, token = Sys.getenv("AHREFS_AUTH_TOKEN"),
  mode = "domain", metrics = NULL, limit = 1000, order_by = NULL,
  where = NULL, having = NULL)`

**Arguments**

- **target**
  - character string. Aim of a request: a domain, a directory or a URL
- **token**
  - character string. Authentication token. Should be available through environmental variables after authentication with function `rah_auth()`
- **mode**
  - character string. Mode of operation: exact, domain, subdomains or prefix. See more in Details section
- **metrics**
  - character vector of columns to select. See more in Details section
- **limit**
  - integer. Number of results to return
- **order_by**
  - character vector of columns to sort on. See more in Details section
- **where**
  - character string - a condition created by `rah_condition_set()` function that generates proper "where" condition to satisfy. See more in Details section
- **having**
  - character string - a condition created by `rah_condition_set()` function that generates proper "having" condition to satisfy. See more in Details section
**rahs_anchors**

**Details**

1. **available metrics** - you can select which columns (metrics) you want to download and which one would be useful in filtering, **BUT not all of them can always be used in "where" & "having" conditions:**

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Where</th>
<th>Having</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>anchor</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Anchor text used in at least one backlink from the referring domain.</td>
</tr>
<tr>
<td>backlinks</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Number of external backlinks found that are using the anchor text.</td>
</tr>
<tr>
<td>refpages</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Number of pages found containing backlinks that are using the anchor text.</td>
</tr>
<tr>
<td>refdomain</td>
<td>string</td>
<td>+</td>
<td>-</td>
<td>Referring domain that contains at least one backlink using the anchor text.</td>
</tr>
<tr>
<td>refdomains</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>Number of referring domains that are using the anchor text when linking to the target.</td>
</tr>
<tr>
<td>first_seen</td>
<td>date</td>
<td>+</td>
<td>+</td>
<td>Least recent date when the Ahrefs crawler was able to visit the backlink that is using the anchor text.</td>
</tr>
<tr>
<td>last_visited</td>
<td>date</td>
<td>+</td>
<td>+</td>
<td>Most recent date when the Ahrefs crawler was able to visit the backlink that is using the anchor text.</td>
</tr>
</tbody>
</table>

2. **"mode" parameter** can take 4 different values that will affect how the results will be grouped.

Example of URL directory with folder:

- **Example URL:** ahrefs.com/api/
  - **exact:** ahrefs.com/api/
  - **domain:** ahrefs.com/*
  - **subdomains:** *ahrefs.com/*
  - **prefix:** ahrefs.com/api/*

Example of URL directory with subdomain:

- **Example URL:** apiv2.ahrefs.com
  - **exact:** apiv2.ahrefs.com/
  - **domain:** apiv2.ahrefs.com/*
  - **subdomains:** *apiv2.ahrefs.com/*
  - **prefix:** apiv2.ahrefs.com/*

3. **"order_by" parameter** is a character string that forces sorting of the results. Structure:

- **Structure:** "column_name:asc|desc"
- **Single column example:** "first_seen:asc" ~ this sorts results by first_seen column in ascending order
- **Multi column example:** "last_seen:desc,first_seen:asc" ~ this sorts results by 1) last_seen column in descending order, and next by 2) first_seen column in ascending order

4. **"where" & "having" are EXPERIMENTAL** parameters of condition sets (character strings) that control filtering the results. To create arguments:

1. use **rah_condition()** function to create a single condition, for example: cond_1 <- rah_condition(column = "links", operator = "GREATER_THAN", value = "10")
2. use **rah_condition_set()** function to group single conditions into final condition string, for example: fin_cond <- rah_condition_set(cond_1, cond_2)
3. provide final condition to proper report function as a parameter, for example: RAhrefs::rah_anchors(target = "ahrefs.com", token = "0123456789", mode = "domain", metrics = NULL, limit = 1000, where = fin_cond, order_by = "first_seen:asc")
Value
data frame

Source

https://ahrefs.com/api/documentation
https://ahrefs.com/api/documentation

See Also

Other Ahrefs reports: rah_ahrefs_rank, rah_anchors_refdomains, rah_backlinks_new_lost_counters, rah_backlinks_new_lost, rah_backlinks_one_per_domain, rah_backlinks, rah_broken_backlinks, rah_broken_links, rah_domain_rating, rah_linked_anchors, rah_linked_domains_by_type, rah_linked_domains, rah_metrics_extended, rah_metrics, rah_pages_extended, rah_pages_info, rah_pages, rah_refdomains_by_type, rah_refdomains_new_lost_counters, rah_refdomains_new_lost, rah_refdomains, rah_refips, rah_subscription_info

Examples

## Not run:
# creating single conditions for 'where' parameter
# let's see anchors of all backlinks detected in 2018
cond_1 <- RAhrefs::rah_condition(
  column_name = "first_seen",
  operator = "GREATER_OR_EQUAL",
  value = "2018-01-01",
  is_date = TRUE)

cond_2 <- RAhrefs::rah_condition(
  column_name = "first_seen",
  operator = "LESS_OR_EQUAL",
  value = "2018-12-31",
  is_date = TRUE)

# joining conditions into one condition set
cond_where <- RAhrefs::rah_condition_set(cond_1, cond_2)

# downloading
b <- RAhrefs::rah_anchors(
  target = "ahrefs.com",
  limit = 2,
  where = cond_where,
  order_by = "refpages:desc")

## End(Not run)
rah_anchors_refdomains

Export connection between anchors and domains. Can be used to get all referring domains with specified anchor.

Description

Export connection between anchors and domains. Can be used to get all referring domains with specified anchor.

Usage

rah_anchors_refdomains(target, token = Sys.getenv("AHREFS_AUTH_TOKEN"), mode = "domain", metrics = NULL, limit = 1000, order_by = NULL, where = NULL, having = NULL)

Arguments

- **target**: character string. Aim of a request: a domain, a directory or a URL
- **token**: character string. Authentication token. Should be available through enviromental variables after authentication with function rah_auth()
- **mode**: character string. Mode of operation: exact, domain, subdomains or prefix. See more in Details section
- **metrics**: character vector of columns to select. See more in Details section
- **limit**: integer. Number of results to return
- **order_by**: character vector of columns to sort on. See more in Details section
- **where**: character string - a condition created by rah_condition_set() function that generates proper "where" condition to satisfy. See more in Details section
- **having**: character string - a condition created by rah_condition_set() function that generates proper "having" condition to satisfy. See more in Details section

Details

1. **available metrics** - you can select which columns (metrics) you want to download and which one would be useful in filtering, **BUT not all of them can always be used** in "where" & "having" conditions:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Where</th>
<th>Having</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>anchor</td>
<td>string</td>
<td>+</td>
<td>-</td>
<td>Anchor text used in at least one backlink from the referring domain.</td>
</tr>
<tr>
<td>anchors</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>Number of different anchor texts that are used in backlinks to the target on the referring domain.</td>
</tr>
<tr>
<td>backlinks</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Number of external backlinks found that are using the anchor text.</td>
</tr>
<tr>
<td>refdomain</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Referring domain that contains at least one backlink using the anchor text.</td>
</tr>
</tbody>
</table>

2. "mode" parameter can take 4 different values that will affect how the results will be grouped.
Example of URL directory with folder:

- **Example URL**: ahrefs.com/api/
- **exact**: ahrefs.com/api/
- **domain**: ahrefs.com/*
- **subdomains**: *ahrefs.com/*
- **prefix**: ahrefs.com/api/*

Example of URL directory with subdomain:

- **Example URL**: apiv2.ahrefs.com
- **exact**: apiv2.ahrefs.com/
- **domain**: apiv2.ahrefs.com/*
- **subdomains**: *apiv2.ahrefs.com/*
- **prefix**: apiv2.ahrefs.com/*

3. "order_by" parameter is a character string that forces sorting of the results. Structure:

- **Structure**: "column_name:asc|desc"
- **Single column example**: "first_seen:asc" ~ this sorts results by first_seen column in ascending order
- **Multi column example**: "last_seen:desc,first_seen:asc" ~ this sorts results by 1) last_seen column in descending order, and next by 2) first_seen column in ascending order

4. "where" & "having" are **EXPERIMENTAL** parameters of condition sets (character strings) that control filtering the results. To create arguments:

1. use `rah_condition()` function to create a single condition, for example: `cond_1 <- rah_condition(column = "links", operator = "GREATER_THAN", value = "10")`
2. use `rah_condition_set()` function to group single conditions into final condition string, for example: `fin_cond <- rah_condition_set(cond_1, cond_2)`
3. provide final condition to proper report function as a parameter, for example: `RAhrefs::rah_anchors_refdomains(target = "ahrefs.com", token = "0123456789", mode = "domain", metrics = NULL, limit = 1000, where = fin_cond, order_by = "first_seen:asc")`

**Value**

data frame

**Source**

https://ahrefs.com/api/documentation
https://ahrefs.com/api/documentation
See Also

Other Ahrefs reports: rah_ahrefs_rank, rah_anchors, rah_backlinks_new_lost_counters, rah_backlinks_new_lost, rah_backlinks_one_per_domain, rah_backlinks, rah_broken_backlinks, rah_broken_links, rah_domain_rating, rah_linked_anchors, rah_linked_domains_by_type, rah_linked_domains, rah_metrics, rah_metrics_extended, rah_metrics, rah_pages, rah_pages_extended, rah_pages_info, rah_pages, rah_refdomains, rah_refdomains_by_type, rah_refdomains_new_lost, rah_refdomains_new_lost, rah_refdomains, rah_refdomains_new_lost, rah_refdomains, rah_refdomains, rah_refips, rah_subscription_info

Examples

```r
## Not run:
# creating single conditions for 'where' parameter
cond_1 <- RAhrefs::rah_condition(
  column_name = "backlinks",
  operator = "GREATER_OR_EQUAL",
  value = "10")

# joining conditions into one condition set
cond_where <- RAhrefs::rah_condition_set(cond_1)

# downloading
b <- RAhrefs::rah_anchors_refdomains(
  target = "ahrefs.com",
  limit = 2,
  where = cond_where,
  order_by = "anchors:desc")

## End(Not run)
```

---

**rah_auth**

*Authorize your Ahrefs API connection with a API Key (Token)*

**Description**

Authorize your Ahrefs API connection with a API Key (Token)

**Usage**

```r
rah_auth(api_key, verbose = TRUE)
```

**Arguments**

- `api_key` character string. Valid API key obtained at: https://ahrefs.com/api/profile
- `verbose` logical, defaults to TRUE. Set to FALSE to stop printing status in the console
Value

invisibly returns API token into environment variable AHREFS_AUTH_TOKEN and prints the status

Examples

```r
## Not run:
rah_auth("ABCDEFGHIJKLMNOPQRSTUVWXYZ")
## End(Not run)
```

rah_backlinks  

Export the backlinks and details of the referring pages, such as anchor and page title.

Description

Export the backlinks and details of the referring pages, such as anchor and page title.

Usage

```r
rah_backlinks(target, token = Sys.getenv("AHREFS_AUTH_TOKEN"),
               mode = "domain", metrics = NULL, limit = 1000, order_by = NULL,
               where = NULL, having = NULL)
```

Arguments

- `target` character string. Aim of a request: a domain, a directory or a URL
- `token` character string. Authentication token. Should be available through enviromental variables after authentication with function `rah_auth()`
- `mode` character string. Mode of operation: exact, domain, subdomains or prefix. See more in Details section
- `metrics` character vector of columns to select. See more in Details section
- `limit` integer. Number of results to return
- `order_by` character vector of columns to sort on. See more in Details section
- `where` character string - a condition created by `rah_condition_set()` function that generates proper "where" condition to satisfy. See more in Details section
- `having` character string - a condition created by `rah_condition_set()` function that generates proper "having" condition to satisfy. See more in Details section

Details

1. available metrics - you can select which columns (metrics) you want to download and which one would be useful in filtering, BUT not all of them can always be used in "where" & "having" conditions:
<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Where</th>
<th>Having</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url_from</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>URL of the page where the backlink is found.</td>
</tr>
<tr>
<td>url_to</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>URL of the page the backlink is pointing to.</td>
</tr>
<tr>
<td>ahrefs_rank</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>URL Rating of the referring page.</td>
</tr>
<tr>
<td>domain_rating</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>Domain Rating of the referring domain.</td>
</tr>
<tr>
<td>ahrefs_top</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>Ahrefs Rank of the target domain.</td>
</tr>
<tr>
<td>ip_from</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>IP address of the referring page.</td>
</tr>
<tr>
<td>links_internal</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Number of internal links found in the referring page.</td>
</tr>
<tr>
<td>links_external</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Number of external links found in the referring page.</td>
</tr>
<tr>
<td>page_size</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Size of the referring page, in bytes.</td>
</tr>
<tr>
<td>encoding</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Character encoding of the referring page, e.g. &quot;utf8&quot; or &quot;iso-8859-1&quot;.</td>
</tr>
<tr>
<td>language</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Language of the referring page (ISO 639-1).</td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Title of the referring page.</td>
</tr>
<tr>
<td>first_seen</td>
<td>date</td>
<td>+</td>
<td>+</td>
<td>Least recent date when the Ahrefs crawler was able to visit the backlink.</td>
</tr>
<tr>
<td>last_visited</td>
<td>date</td>
<td>+</td>
<td>+</td>
<td>Most recent date when the Ahrefs crawler was able to visit the backlink.</td>
</tr>
<tr>
<td>prev_visited</td>
<td>date</td>
<td>+</td>
<td>+</td>
<td>Second to the most recent date when the Ahrefs crawler was able to visit the backlink.</td>
</tr>
<tr>
<td>original</td>
<td>boolean</td>
<td>+</td>
<td>+</td>
<td>Indicates whether the backlink was present on the referring page when the Ahrefs crawler first visited it.</td>
</tr>
<tr>
<td>link_type</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Either &quot;href&quot;, &quot;redirect&quot;, &quot;frame&quot;, &quot;form&quot;, &quot;canonical&quot;, &quot;rss&quot;, or &quot;alternate&quot;.</td>
</tr>
<tr>
<td>redirect</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>For redirected links, the Redirect Code (3XX), zero otherwise.</td>
</tr>
<tr>
<td>nolofollow</td>
<td>boolean</td>
<td>+</td>
<td>+</td>
<td>Indicates whether the backlink is NoFollow.</td>
</tr>
<tr>
<td>alt</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Alternative text of the image backlink, if exists.</td>
</tr>
<tr>
<td>anchor</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Anchor text of the backlink.</td>
</tr>
<tr>
<td>text_pre</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Snippet before the anchor text.</td>
</tr>
<tr>
<td>text_post</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Snippet after the anchor text.</td>
</tr>
<tr>
<td>http_code</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>The HTTP code for the Link URL.</td>
</tr>
<tr>
<td>url_from_first_seen</td>
<td>date</td>
<td>+</td>
<td>+</td>
<td>Least recent date when the Ahrefs crawler was able to visit the referring page with backlink.</td>
</tr>
</tbody>
</table>

2. "mode" parameter can take 4 different values that will affect how the results will be grouped.

Example of URL directory with folder:

- **Example URL**: ahrefs.com/api/
- **exact**: ahrefs.com/api/
- **domain**: ahrefs.com/*
- **subdomains**: *ahrefs.com/*
- **prefix**: ahrefs.com/api/*

Example of URL directory with subdomain:

- **Example URL**: apiv2.ahrefs.com
- **exact**: apiv2.ahrefs.com/
- **domain**: apiv2.ahrefs.com/*
- **subdomains**: *apiv2.ahrefs.com/*
- **prefix**: apiv2.ahrefs.com/*

3. "order_by" parameter is a character string that forces sorting of the results. Structure:

- **Structure**: "column_name:asc|desc"
• **Single column example**: "first_seen:asc" ~ this sorts results by first_seen column in ascending order

• **Multi column example**: "last_seen:desc,first_seen:asc" ~ this sorts results by 1) last_seen column in descending order, and next by 2) first_seen column in ascending order

4. "where" & "having" are **EXPERIMENTAL** parameters of condition sets (character strings) that control filtering the results. To create arguments:

1. use `rah_condition()` function to create a single condition, for example: `cond_1 <- rah_condition(column = "links", operator = "GREATER_THAN", value = "10")`

2. use `rah_condition_set()` function to group single conditions into final condition string, for example: `fin_cond <- rah_condition_set(cond_1, cond_2)`

3. provide final condition to proper report function as a parameter, for example: `RAhrefs::rah_backlinks(target = "ahrefs.com", token = "0123456789", mode = "domain", metrics = NULL, limit = 1000, where = fin_cond, order_by = "first_seen:asc")`

**Value**

data frame

**Source**

https://ahrefs.com/api/documentation

**See Also**

Other Ahrefs reports: rah_ahrefs_rank, rah_anchors_refdomains, rah_anchors, rah_backlinks_new_lost_counters, rah_backlinks_new_lost, rah_backlinks_one_per_domain, rah_broken_backlinks, rah_broken_links, rah_domain_rating, rah_linked_anchors, rah_linked_domains_by_type, rah_linked_domains, rah_metrics_extended, rah_metrics, rah_pages_extended, rah_pages_info, rah_pages, rah_refdomains_by_type, rah_refdomains_new_lost_counters, rah_refdomains_new_lost, rah_refdomains, rah_refips, rah_subscription_info

**Examples**

```r
## Not run:
# creating single conditions for 'where' parameter
cond_1 <- RAhrefs::rah_condition(
  column_name = "ahrefs_rank",
  operator = "GREATER_OR_EQUAL",
  value = "20")

# joining conditions into one condition set
cond_where <- RAhrefs::rah_condition_set(cond_1)

# downloading
b <- RAhrefs::rah_backlinks(
  target = "ahrefs.com",
  limit = 2,
)```
**rah_backlinks_new_lost**

Export the new or lost backlinks and details of the referring pages.

**Description**

Export the new or lost backlinks and details of the referring pages.

**Usage**

```r
rah_backlinks_new_lost(target, token = Sys.getenv("AHREFS_AUTH_TOKEN"),
    mode = "domain", metrics = NULL, limit = 1000, order_by = NULL,
    where = NULL, having = NULL)
```

**Arguments**

- **target** character string. Aim of a request: a domain, a directory or a URL
- **token** character string. Authentication token. Should be available through enviromental variables after authentication with function `rah_auth()`
- **mode** character string. Mode of operation: exact, domain, subdomains or prefix. See more in Details section
- **metrics** character vector of columns to select. See more in Details section
- **limit** integer. Number of results to return
- **order_by** character vector of columns to sort on. See more in Details section
- **where** character string - a condition created by `rah_condition_set()` function that generates proper "where" condition to satisfy. See more in Details section
- **having** character string - a condition created by `rah_condition_set()` function that generates proper "having" condition to satisfy. See more in Details section

**Details**

1. **available metrics** - you can select which columns (metrics) you want to download and which one would be useful in filtering, **BUT not all of them can always be used** in "where" & "having" conditions:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Where</th>
<th>Having</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>date</td>
<td>date</td>
<td>+</td>
<td>+</td>
<td>Date the backlink was tagged as New or Lost.</td>
</tr>
<tr>
<td>type</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Indicates whether the backlink is New or Lost.</td>
</tr>
<tr>
<td>ahrefs_rank</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>URL Rating of the referring page.</td>
</tr>
<tr>
<td>domain_rating</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Domain Rating of the referring domain.</td>
</tr>
</tbody>
</table>
2. "mode" parameter can take 4 different values that will affect how the results will be grouped.

Example of URL directory with folder:

- Example URL: ahrefs.com/api/
- exact: ahrefs.com/api/
- domain: ahrefs.com/*
- subdomains: *ahrefs.com/*
- prefix: ahrefs.com/api/*

Example of URL directory with subdomain:

- Example URL: apiv2.ahrefs.com
- exact: apiv2.ahrefs.com/
- domain: apiv2.ahrefs.com/*
- subdomains: *apiv2.ahrefs.com/*
- prefix: apiv2.ahrefs.com/*

3. "order_by" parameter is a character string that forces sorting of the results. Structure:

- Structure: "column_name:asc|desc"
- Single column example: "first_seen:asc" ~ this sorts results by first_seen column in ascending order
- Multi column example: "last_seen:desc,first_seen:asc" ~ this sorts results by 1) last_seen column in descending order, and next by 2) first_seen column in ascending order

4. "where" & "having" are EXPERIMENTAL parameters of condition sets (character strings) that control filtering the results. To create arguments:
1. use `rah_condition()` function to create a single condition, for example: `cond_1 <- rah_condition(column = "links", operator = "GREATER_THAN", value = "10")`

2. use `rah_condition_set()` function to group single conditions into final condition string, for example: `fin_cond <- rah_condition_set(cond_1, cond_2)`

3. provide final condition to proper report function as a parameter, for example: RAhrefs::rah_backlinks_new_lost(target = "ahrefs.com", token = "0123456789", mode = "domain", metrics = NULL, limit = 1000, where = fin_cond, order_by = "first_seen:asc")

Value
data frame

Source
https://ahrefs.com/api/documentation
https://ahrefs.com/api/documentation

See Also
Other Ahrefs reports: `rah_ahrefs_rank`, `rah_anchors_refdomains`, `rah_anchors`, `rah_backlinks_new_lost_counters`, `rah_backlinks_one_per_domain`, `rah_backlinks`, `rah_broken_backlinks`, `rah_broken_links`, `rah_domain_rating`, `rah_linkedin_domain`, `rah_linked_domains_by_type`, `rah_linked_domains`, `rah_metrics`, `rah_metrics_extended`, `rah_pages_extended`, `rah_pages_info`, `rah_pages`, `rah_refdomains_by_type`, `rah_refdomains_new_lost_counters`, `rah_refdomains_new_lost`, `rah_refdomains`, `rah_refips`, `rah_subscription_info`

Examples
## Not run:
# creating single conditions for 'where' parameter
# let's see anchors of all backlinks detected in 2018
cond_1 <- RAhrefs::rah_condition(
  column_name = "nofollow",
  operator = "EQUALS",
  value = 0)

cond_2 <- RAhrefs::rah_condition(
  column_name = "last_visited",
  operator = "LESS_OR_EQUAL",
  value = "2018-05-31",
  is_date = TRUE)

# joining conditions into one condition set
cond_where <- RAhrefs::rah_condition_set(cond_1, cond_2)

# downloading
b <- RAhrefs::rah_backlinks_new_lost(
  target = "ahrefs.com",
  limit = 2,
rah_backlinks_new_lost_counters

Export new and lost backlink totals.

Description

Export new and lost backlink totals.

Usage

```r
rah_backlinks_new_lost_counters(target,
  token = Sys.getenv("AHREFS_AUTH_TOKEN"), mode = "domain",
  metrics = NULL, limit = 1000, order_by = NULL, where = NULL,
  having = NULL)
```

Arguments

- **target** character string. Aim of a request: a domain, a directory or a URL
- **token** character string. Authentication token. Should be available through enviromental variables after authentication with function `rah_auth()`
- **mode** character string. Mode of operation: exact, domain, subdomains or prefix. See more in Details section
- **metrics** character vector of columns to select. See more in Details section
- **limit** integer. Number of results to return
- **order_by** character vector of columns to sort on. See more in Details section
- **where** character string - a condition created by `rah_condition_set()` function that generates proper "where" condition to satisfy. See more in Details section
- **having** character string - a condition created by `rah_condition_set()` function that generates proper "having" condition to satisfy. See more in Details section

Details

1. **available metrics** - you can select which columns (metrics) you want to download and which one would be useful in filtering, **BUT not all of them can always be used** in "where" & "having" conditions:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Where</th>
<th>Having</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>date</td>
<td>date</td>
<td>+</td>
<td>+</td>
<td>Date the backlink was tagged as New or Lost</td>
</tr>
<tr>
<td>type</td>
<td>string</td>
<td>+</td>
<td>-</td>
<td>Indicates whether the backlink is New or Lost.</td>
</tr>
<tr>
<td>ahrefs_rank</td>
<td>int</td>
<td>+</td>
<td>-</td>
<td>URL Rating of the referring page.</td>
</tr>
</tbody>
</table>
domain_rating int + - Domain Rating of the referring domain.
url_from string + - URL of the page where the backlink is found.
url_to string + + URL of the page the backlink is pointing to.
links_internal int + - Number of internal links found in the referring page.
links_external int + - Number of external links found in the referring page.
encoding string + - Character encoding of the referring page, for example "utf8" or "iso-8859-1" (Latin-1).
http_code int + - HTTP code that was last returned for the referring page.
title string + - Title of the referring page.
origin string + - Either "fresh", "drop", or "recrawl".
first_seen date + - Least recent date when the Ahrefs crawler was able to visit the backlink.
last_visited date + - Most recent date when the Ahrefs crawler was able to visit the backlink.
prev_visited date + - Second to most recent date when the Ahrefs crawler was able to visit the backlink.
original boolean + - Indicates whether the backlink was present on the referring page when the Ahrefs crawler first visited it.
link_type string + - Either "href", "redirect", "frame", "form", "canonical", "rss", or "alternate".
redirect int + - For redirected links, the Redirect Code (3XX), zero otherwise.
nofollow boolean + - Indicates whether the backlink is NoFollow.
at string + - Alternative text of the image backlink, if exists.
anchor string + - Anchor text of the backlink.
text_pre string + - Snippet before the anchor text.
text_post string + - Snippet after the anchor text.
new int - + Total number of new backlinks found to this url.
lost int - + Total number of backlinks removed to this url.
new_total int - + Total number of new backlinks found to this url when ignoring where filter.
lost_total int - + Total number of backlinks removed to this url when ignoring where filter.

2. "mode" parameter can take 4 different values that will affect how the results will be grouped.
Example of URL directory with folder:

- **Example URL:** ahrefs.com/api/
- **exact:** ahrefs.com/api/
- **domain:** ahrefs.com/*
- **subdomains:** *ahrefs.com/*
- **prefix:** ahrefs.com/api/*

Example of URL directory with subdomain:

- **Example URL:** apiv2.ahrefs.com
- **exact:** apiv2.ahrefs.com/
- **domain:** apiv2.ahrefs.com/*
- **subdomains:** *apiv2.ahrefs.com/*
- **prefix:** apiv2.ahrefs.com/*

3. "order_by" parameter is a character string that forces sorting of the results. Structure:

- **Structure:** "column_name:asc|desc"
- **Single column example:** "first_seen:asc" ~ this sorts results by first_seen column in ascending order
- **Multi column example**: "last_seen:desc,first_seen:asc" - this sorts results by 1) last_seen column in descending order, and next by 2) first_seen column in ascending order

4. "where" & "having" are **EXPERIMENTAL** parameters of condition sets (character strings) that control filtering the results. To create arguments:

1. use `rah_condition()` function to create a single condition, for example: `cond_1 <- rah_condition(column = "links", operator = "GREATER_THAN", value = "10")`
2. use `rah_condition_set()` function to group single conditions into final condition string, for example: `fin_cond <- rah_condition_set(cond_1, cond_2)`
3. provide final condition to proper report function as a parameter, for example: `RAhrefs::rah_backlinks_new_lost_counters(target = "ahrefs.com", token = "0123456789", mode = "domain", metrics = NULL, limit = 1000, where = fin_cond, order_by = "first_seen:asc")`

**Value**

data frame

**Source**

https://ahrefs.com/api/documentation

**See Also**

Other Ahrefs reports: `rah_ahrefs_rank`, `rah_anchors_refdomains`, `rah_anchors`, `rah_backlinks_new_lost`, `rah_backlinks_one_per_domain`, `rah_backlinks`, `rah_broken_backlinks`, `rah_broken_links`, `rah_domain_rating`, `rah_linked_anchors`, `rah_linked_domains_by_type`, `rah_linked_domains`, `rah_metrics`, `rah_metrics_extended`, `rah_pages`, `rah_pages_extended`, `rah_pages_info`, `rah_refdomains`, `rah_refdomains_by_type`, `rah_refdomains_new_lost`, `rah_refdomains`, `rah_refips`, `rah_subscription_info`

**Examples**

```r
## Not run:
# creating single conditions for 'where' parameter
# let's see anchors of all backlinks detected in 2018
cond_1 <- RAhrefs::rah_condition(
  column_name = "links_internal",
  operator = "GREATER_OR_EQUAL",
  value = "5")

cond_2 <- RAhrefs::rah_condition(
  column_name = "links_external",
  operator = "LESS_OR_EQUAL",
  value = "5")

# joining conditions into one condition set
cond_where <- RAhrefs::rah_condition_set(cond_1, cond_2)

# downloading
```
rah_backlinks_one_per_domain

Export the backlinks and details of the referring pages, such as anchor and page title.

Description

Export the backlinks and details of the referring pages, such as anchor and page title.

Usage

rah_backlinks_one_per_domain(target, 
	token = Sys.getenv("AHREFS_AUTH_TOKEN"), mode = "domain", 
metrics = NULL, limit = 1000, order_by = NULL, where = NULL, 
having = NULL)

Arguments

target character string. Aim of a request: a domain, a directory or a URL

token character string. Authentication token. Should be available through enviromental variables after authentication with function rah_auth()

mode character string. Mode of operation: exact, domain, subdomains or prefix. See more in Details section

metrics character vector of columns to select. See more in Details section

limit integer. Number of results to return

order_by character vector of columns to sort on. See more in Details section

where character string - a condition created by rah_condition_set() function that generates proper "where" condition to satisfy. See more in Details section

having character string - a condition created by rah_condition_set() function that generates proper "having" condition to satisfy. See more in Details section

Details

1. available metrics - you can select which columns (metrics) you want to download and which one would be useful in filtering, **BUT not all of them can always be used** in "where" & "having" conditions:
<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Where</th>
<th>Having</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url_from</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>URL of the page where the backlink is found.</td>
</tr>
<tr>
<td>url_to</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>URL of the page the backlink is pointing to.</td>
</tr>
<tr>
<td>ahrefs_rank</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>URL Rating of the referring page.</td>
</tr>
<tr>
<td>domain_rating</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>Domain Rating of the referring domain.</td>
</tr>
<tr>
<td>ahrefs_top</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>Ahrefs Rank of the target domain.</td>
</tr>
<tr>
<td>ip_from</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>IP address of the referring page.</td>
</tr>
<tr>
<td>links_internal</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Number of internal links found in the referring page.</td>
</tr>
<tr>
<td>links_external</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Number of external links found in the referring page.</td>
</tr>
<tr>
<td>page_size</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Size of the referring page, in bytes.</td>
</tr>
<tr>
<td>encoding</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Character encoding of the referring page, for example &quot;utf8&quot; or &quot;iso-8859-1&quot;</td>
</tr>
<tr>
<td>language</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Language of the referring page (ISO 639-1).</td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Title of the referring page.</td>
</tr>
<tr>
<td>first_seen</td>
<td>date</td>
<td>+</td>
<td>+</td>
<td>Least recent date when the Ahrefs crawler was able to visit the backlink.</td>
</tr>
<tr>
<td>last_visited</td>
<td>date</td>
<td>+</td>
<td>+</td>
<td>Most recent date when the Ahrefs crawler was able to visit the backlink.</td>
</tr>
<tr>
<td>prev_visited</td>
<td>date</td>
<td>+</td>
<td>+</td>
<td>Second to the most recent date when the Ahrefs crawler was able to visit.</td>
</tr>
<tr>
<td>original</td>
<td>boolean</td>
<td>+</td>
<td>+</td>
<td>Indicates whether the backlink was present on the referring page when the Ahrefs crawler first visited it.</td>
</tr>
<tr>
<td>link_type</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Either &quot;href&quot;, &quot;redirect&quot;, &quot;frame&quot;, &quot;form&quot;, &quot;canonical&quot;, &quot;rss&quot;, or &quot;alternate&quot;</td>
</tr>
<tr>
<td>redirect</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>For redirected links, the Redirect Code (3XX), zero otherwise.</td>
</tr>
<tr>
<td>nofollow</td>
<td>boolean</td>
<td>+</td>
<td>+</td>
<td>Indicates whether the backlink is NoFollow.</td>
</tr>
<tr>
<td>alt</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Alternative text of the image backlink, if exists.</td>
</tr>
<tr>
<td>anchor</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Anchor text of the backlink.</td>
</tr>
<tr>
<td>text_pre</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Snippet before the anchor text.</td>
</tr>
<tr>
<td>text_post</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Snippet after the anchor text.</td>
</tr>
<tr>
<td>http_code</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>The HTTP code for the Link URL.</td>
</tr>
<tr>
<td>url_from_first_seen</td>
<td>date</td>
<td>+</td>
<td>+</td>
<td>Least recent date when the Ahrefs crawler was able to visit the referring page with backlink.</td>
</tr>
<tr>
<td>total_backlinks</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>Total number of backlinks from this referring domain.</td>
</tr>
</tbody>
</table>

2. "mode" parameter can take 4 different values that will affect how the results will be grouped.

Example of URL directory with folder:

- **Example URL**: ahrefs.com/api/
- **exact**: ahrefs.com/api/
- **domain**: ahrefs.com/*
- **subdomains**: *ahrefs.com/*
- **prefix**: ahrefs.com/api/*

Example of URL directory with subdomain:

- **Example URL**: apiv2.ahrefs.com
- **exact**: apiv2.ahrefs.com/
- **domain**: apiv2.ahrefs.com/*
- **subdomains**: *apiv2.ahrefs.com/*
- **prefix**: apiv2.ahrefs.com/*

3. "order_by" parameter is a character string that forces sorting of the results. Structure:
• Structure: "column_name:asc|desc"
• Single column example: "first_seen:asc" – this sorts results by first_seen column in ascending order
• Multi column example: "last_seen:desc,first_seen:asc" – this sorts results by 1) last_seen column in descending order, and next by 2) first_seen column in ascending order

4. "where" & "having" are EXPERIMENTAL parameters of condition sets (character strings) that control filtering the results. To create arguments:

1. use `rah_condition()` function to create a single condition, for example: `cond_1 <- rah_condition(column_name = "first_seen", operator = "GREATER_THAN", value = "10")`
2. use `rah_condition_set()` function to group single conditions into final condition string, for example: `fin_cond <- rah_condition_set(cond_1, cond_2)`
3. provide final condition to proper report function as a parameter, for example: `RAhrefs::rah_backlinks_one_per_domain(target = "ahrefs.com", token = "0123456789", mode = "domain", metrics = NULL, limit = 1000, where = fin_cond, order_by = "first_seen:asc")`

Value
data frame

Source
https://ahrefs.com/api/documentation

See Also
Other Ahrefs reports: `rah_ahrefs_rank`, `rah_anchors_refdomains`, `rah_anchors`, `rah_backlinks_new_lost_counters`, `rah_backlinks_new_lost`, `rah_backlinks`, `rah_broken_backlinks`, `rah_broken_links`, `rah_domain_rating`, `rah_linked_anchors`, `rah_linked_domains_by_type`, `rah_linked_domains`, `rah_metrics`, `rah_metrics_extended`, `rah_metrics`, `rah_pages_extended`, `rah_pages_info`, `rah_pages`, `rah_refdomains_by_type`, `rah_refdomains_new_lost_counters`, `rah_refdomains_new_lost`, `rah_refdomains`, `rah_refips`, `rah_subscription_info`

Examples

```r
## Not run:
# creating single conditions for 'where' parameter
cond_1 <- RAhrefs::rah_condition(
  column_name = "first_seen",
  operator    = "GREATER_OR_EQUAL",
  value       = "2018-01-01",
  is_date     = TRUE)

cond_2 <- RAhrefs::rah_condition(
  column_name = "page_size",
  operator    = "LESS_OR_EQUAL",
  value       = "2048000")
```
# joining conditions into one condition set
cond_where <- RAhrefs::rah_condition_set(cond_1, cond_2)

# downloading
b <- RAhrefs::rah_backlinks_one_per_domain(target = "ahrefs.com",
                                          limit = 2,
                                          where = cond_where,
                                          order_by = "ahrefs_rank:desc")

## End(Not run)

### Description

Export the broken backlinks and details of the referring pages, such as anchor and page title.

### Usage

```
rah_broken_backlinks(target, token = Sys.getenv("AHREFS_AUTH_TOKEN"),
                      mode = "domain", metrics = NULL, limit = 1000, order_by = NULL,
                      where = NULL, having = NULL)
```

### Arguments

- **target**: character string. Aim of a request: a domain, a directory or a URL.
- **token**: character string. Authentication token. Should be available through environmental variables after authentication with function `rah_auth()`.
- **mode**: character string. Mode of operation: exact, domain, subdomains or prefix. See more in Details section.
- **metrics**: character vector of columns to select. See more in Details section.
- **limit**: integer. Number of results to return.
- **order_by**: character vector of columns to sort on. See more in Details section.
- **where**: character string - a condition created by `rah_condition_set()` function that generates proper "where" condition to satisfy. See more in Details section.
- **having**: character string - a condition created by `rah_condition_set()` function that generates proper "having" condition to satisfy. See more in Details section.

### Details

1. **available metrics** - you can select which columns (metrics) you want to download and which one would be useful in filtering, **BUT not all of them can always be used** in "where" & "having" conditions:
<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Where</th>
<th>Having</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url_from</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>URL of the page where the backlink is found.</td>
</tr>
<tr>
<td>url_to</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>URL of the page the backlink is pointing to.</td>
</tr>
<tr>
<td>ahrefs_rank</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>URL Rating of the referring page.</td>
</tr>
<tr>
<td>domain_rating</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>Domain Rating of the referring domain.</td>
</tr>
<tr>
<td>ip_from</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>IP address of the referring page.</td>
</tr>
<tr>
<td>links_internal</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Number of internal links found in the referring page.</td>
</tr>
<tr>
<td>links_external</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Number of external links found in the referring page.</td>
</tr>
<tr>
<td>page_size</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Size of the referring page, in bytes.</td>
</tr>
<tr>
<td>language</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Language of the referring page (ISO 639-1).</td>
</tr>
<tr>
<td>encoding</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Character encoding of the referring page, for example &quot;utf8&quot; or &quot;iso-8859-1&quot; (Latin-1).</td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Title of the referring page.</td>
</tr>
<tr>
<td>first_seen</td>
<td>date</td>
<td>+</td>
<td>+</td>
<td>Least recent date when the Ahrefs crawler was able to visit the backlink.</td>
</tr>
<tr>
<td>last_visited</td>
<td>date</td>
<td>+</td>
<td>+</td>
<td>Most recent date when the Ahrefs crawler was able to visit the backlink.</td>
</tr>
<tr>
<td>prev_visited</td>
<td>date</td>
<td>+</td>
<td>+</td>
<td>Second to the most recent date when the Ahrefs crawler was able to visit the backlink.</td>
</tr>
<tr>
<td>original</td>
<td>boolean</td>
<td>+</td>
<td>+</td>
<td>Indicates whether the backlink was present on the referring page when the Ahrefs crawler first visited it.</td>
</tr>
<tr>
<td>link_type</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Either &quot;href&quot;, &quot;redirect&quot;, &quot;frame&quot;, &quot;form&quot;, &quot;canonical&quot;, &quot;rss&quot;, or &quot;alternate&quot;.</td>
</tr>
<tr>
<td>redirect</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>For redirected links, the Redirect Code (3XX), zero otherwise.</td>
</tr>
<tr>
<td>nofollow</td>
<td>boolean</td>
<td>+</td>
<td>+</td>
<td>Indicates whether the backlink is NoFollow.</td>
</tr>
<tr>
<td>alt</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Alternative text of the image backlink, if exists.</td>
</tr>
<tr>
<td>anchor</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Anchor text of the backlink.</td>
</tr>
<tr>
<td>text_pre</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Snippet before the anchor text.</td>
</tr>
<tr>
<td>text_post</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Snippet after the anchor text.</td>
</tr>
<tr>
<td>broken_at</td>
<td>date</td>
<td>+</td>
<td>+</td>
<td>The date when the Ahrefs crawler thinks the link became broken.</td>
</tr>
<tr>
<td>http_code</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>The HTTP code for the Link URL.</td>
</tr>
<tr>
<td>error</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>The string indicating the nature of error (currently only &quot;dns&quot;).</td>
</tr>
</tbody>
</table>

2. "mode" parameter can take 4 different values that will affect how the results will be grouped.

Example of URL directory with folder:

- **Example URL**: ahrefs.com/api/
  - **exact**: ahrefs.com/api/
  - **domain**: ahrefs.com/*
  - **subdomains**: *ahrefs.com/*
  - **prefix**: ahrefs.com/api/*

Example of URL directory with subdomain:

- **Example URL**: apiv2.ahrefs.com
  - **exact**: apiv2.ahrefs.com
  - **domain**: apiv2.ahrefs.com/
  - **subdomains**: *apiv2.ahrefs.com/*
  - **prefix**: apiv2.ahrefs.com/*

3. "order_by" parameter is a character string that forces sorting of the results. Structure:

- **Structure**: "column_name:asc|desc"
• **Single column example**: "first_seen:asc" - this sorts results by `first_seen` column in ascending order

• **Multi column example**: "last_seen:desc,first_seen:asc" - this sorts results by 1) `last_seen` column in descending order, and next by 2) `first_seen` column in ascending order

4. "where" & "having" are **EXPERIMENTAL** parameters of condition sets (character strings) that control filtering the results. To create arguments:

1. use `rah_condition()` function to create a single condition, for example: `cond_1 <- RAhrefs::rah_condition(column = "links", operator = "GREATER_THAN", value = "10")`
2. use `rah_condition_set()` function to group single conditions into final condition string, for example: `fin_cond <- RAhrefs::rah_condition_set(cond_1, cond_2)`
3. provide final condition to proper report function as a parameter, for example: `RAhrefs::rah_broken_backlinks(target = "ahrefs.com", token = "0123456789", mode = "domain", metrics = NULL, limit = 1000, where = fin_cond, order_by = "first_seen:asc")`

**Value**

data frame

**Source**

https://ahrefs.com/api/documentation
https://ahrefs.com/api/documentation

**See Also**

Other Ahrefs reports: `rah_ahrefs_rank`, `rah_anchoers_refdomains`, `rah_anchors`, `rah_backlinks_new_lost_counters`, `rah_backlinks_new_lost`, `rah_backlinks_one_per_domain`, `rah_backlinks`, `rah_broken_links`, `rah_domain_rating`, `rah_linked_anchors`, `rah_linked_domains_by_type`, `rah_linked_domains`, `rah_metrics_extended`, `rah_metrics`, `rah_pages_extended`, `rah_pages_info`, `rah_pages`, `rah_refdomains_by_type`, `rah_refdomains_new_lost_counters`, `rah_refdomains_new_lost`, `rah_refdomains`, `rah_refips`, `rah_subscription_info`

**Examples**

```r
## Not run:
# creating single conditions for 'where' parameter
cond_1 <- RAhrefs::rah_condition(
  column_name = "first_seen",
  operator = "GREATER_OR_EQUAL",
  value = "2018-01-01",
  is_date = TRUE)

cond_2 <- RAhrefs::rah_condition(
  column_name = "http_code",
  operator = "EQUALS",
  value = "404")
```
# joining conditions into one condition set
cond_where <- RAhrefs::rah_condition_set(cond_1, cond_2)

# downloading
b <- RAhrefs::rah_broken_backlinks(
  target = "ahrefs.com",
  limit  = 2,
  where  = cond_where,
  order_by = "refpages:desc")

## End(Not run)

**rah_broken_links**

Export the broken links and details of the referring pages, such as anchor and page title.

### Description

Export the broken links and details of the referring pages, such as anchor and page title.

### Usage

```r
rah_broken_links(target, token = Sys.getenv("AHREFS_AUTH_TOKEN"),
                  mode = "domain", metrics = NULL, limit = 1000, order_by = NULL,
                  where = NULL, having = NULL)
```

### Arguments

- `target` character string. Aim of a request: a domain, a directory or a URL.
- `token` character string. Authentication token. Should be available through enviromental variables after authentication with function `rah_auth()`.
- `mode` character string. Mode of operation: exact, domain, subdomains or prefix. See more in Details section.
- `metrics` character vector of columns to select. See more in Details section.
- `limit` integer. Number of results to return.
- `order_by` character vector of columns to sort on. See more in Details section.
- `where` character string - a condition created by `rah_condition_set()` function that generates proper "where" condition to satisfy. See more in Details section.
- `having` character string - a condition created by `rah_condition_set()` function that generates proper "having" condition to satisfy. See more in Details section.

### Details

1. **available metrics** - you can select which columns (metrics) you want to download and which one would be useful in filtering, **BUT not all of them can always be used** in "where" & "having" conditions:
<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Where</th>
<th>Having</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url_from</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>URL of the page where the backlink is found.</td>
</tr>
<tr>
<td>url_to</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>URL of the page the backlink is pointing to.</td>
</tr>
<tr>
<td>ahrefs_rank</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>URL Rating of the referring page.</td>
</tr>
<tr>
<td>domain_rating</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>Domain Rating of the referring domain.</td>
</tr>
<tr>
<td>ip_from</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>IP address of the referring page.</td>
</tr>
<tr>
<td>links_internal</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Number of internal links found in the referring page.</td>
</tr>
<tr>
<td>links_external</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Number of external links found in the referring page.</td>
</tr>
<tr>
<td>page_size</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Size of the referring page, in bytes.</td>
</tr>
<tr>
<td>encoding</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Character encoding of the referring page, for example &quot;utf8&quot; or &quot;iso-8859-1&quot;.</td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Title of the referring page.</td>
</tr>
<tr>
<td>language</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Language of the referring page (ISO 639-1).</td>
</tr>
<tr>
<td>first_seen</td>
<td>date</td>
<td>+</td>
<td>+</td>
<td>Least recent date when the Ahrefs crawler was able to visit the backlink.</td>
</tr>
<tr>
<td>last_visited</td>
<td>date</td>
<td>+</td>
<td>+</td>
<td>Most recent date when the Ahrefs crawler was able to visit the backlink.</td>
</tr>
<tr>
<td>prev_visited</td>
<td>date</td>
<td>+</td>
<td>+</td>
<td>Second to the most recent date when the Ahrefs crawler was able to visit the backlink.</td>
</tr>
<tr>
<td>original</td>
<td>boolean</td>
<td>+</td>
<td>+</td>
<td>Indicates whether the backlink was present on the referring page when the Ahrefs crawler first visited it.</td>
</tr>
<tr>
<td>link_type</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Either &quot;href&quot;, &quot;redirect&quot;, &quot;frame&quot;, &quot;form&quot;, &quot;canonical&quot;, &quot;rss&quot;, or &quot;alternate&quot;.</td>
</tr>
<tr>
<td>redirect</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>For redirected links, the Redirect Code (3XX), zero otherwise.</td>
</tr>
<tr>
<td>nofollow</td>
<td>boolean</td>
<td>+</td>
<td>+</td>
<td>Indicates whether the backlink is NoFollow.</td>
</tr>
<tr>
<td>alt</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Alternative text of the image backlink, if exists.</td>
</tr>
<tr>
<td>anchor</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Anchor text of the backlink.</td>
</tr>
<tr>
<td>text_pre</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Snippet before the anchor text.</td>
</tr>
<tr>
<td>text_post</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Snippet after the anchor text.</td>
</tr>
<tr>
<td>broken_at</td>
<td>date</td>
<td>+</td>
<td>+</td>
<td>The date when the Ahrefs crawler thinks the link became broken.</td>
</tr>
<tr>
<td>http_code</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>The HTTP code for the Link URL.</td>
</tr>
<tr>
<td>error</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>The string indicating the nature of error (currently only &quot;dns&quot;).</td>
</tr>
<tr>
<td>domain_to_ahrefs_top</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>Ahrefs rank of the external domain.</td>
</tr>
<tr>
<td>url_from_first_seen</td>
<td>date</td>
<td>+</td>
<td>+</td>
<td>Least recent date when the Ahrefs crawler was able to visit the referring page.</td>
</tr>
</tbody>
</table>

2. "mode" parameter can take 4 different values that will affect how the results will be grouped.

Example of URL directory with folder:

- **Example URL**: ahrefs.com/api/
  - **exact**: ahrefs.com/api/
  - **domain**: ahrefs.com/*
  - **subdomains**: *ahrefs.com/*
  - **prefix**: ahrefs.com/api/*

Example of URL directory with subdomain:

- **Example URL**: apiv2.ahrefs.com
  - **exact**: apiv2.ahrefs.com/
  - **domain**: apiv2.ahrefs.com/*
  - **subdomains**: *apiv2.ahrefs.com/*
  - **prefix**: apiv2.ahrefs.com/*

3. "order_by" parameter is a character string that forces sorting of the results. Structure:
• **Structure:** "column_name:asc|desc"

• **Single column example:** "first_seen:asc" ~ this sorts results by first_seen column in ascending order

• **Multi column example:** "last_seen:desc,first_seen:asc" ~ this sorts results by 1) last_seen column in descending order, and next by 2) first_seen column in ascending order

4. "where" & "having" are EXPERIMENTAL parameters of condition sets (character strings) that control filtering the results. To create arguments:

   1. use `rah_condition()` function to create a single condition, for example: `cond_1 < RAhrefs::rah_condition(column = "links", operator = "GREATER_THAN", value = "10")`

   2. use `rah_condition_set()` function to group single conditions into final condition string, for example: `fin_cond < RAhrefs::rah_condition_set(cond_1, cond_2)`

   3. provide final condition to proper report function as a parameter, for example: `RAhrefs::rah_broken_links(target = "ahrefs.com", token = "0123456789", mode = "domain", metrics = NULL, limit = 1000, where = fin_cond, order_by = "first_seen:asc")`

**Value**

data frame

**Source**

https://ahrefs.com/api/documentation

https://ahrefs.com/api/documentation

**See Also**

Other Ahrefs reports: `rah_ahrefs_rank`, `rah_anchors_refdomains`, `rah_anchors`, `rah_backlinks_new_lost_counters`, `rah_backlinks_new_lost`, `rah_backlinks_one_per_domain`, `rah_backlinks`, `rah_broken_backlinks`, `rah_domain_rating`, `rah_linked_anchors`, `rah_linked_domains_by_type`, `rah_linked_domains`, `rah_metrics_extended`, `rah_metrics`, `rah_pages_extended`, `rah_pages_info`, `rah_pages`, `rah_refdomains_by_type`, `rah_refdomains_new_lost_counts`, `rah_refdomains_new_lost`, `rah_refdomains`, `rah_refips`, `rah_subscription_info`

**Examples**

```r
## Not run:
# creating single conditions for 'where' parameter
cond_1 <- RAhrefs::rah_condition(
  column_name = "first_seen",
  operator = "GREATER_OR_EQUAL",
  value = "2018-01-01",
  is_date = TRUE)

cond_2 <- RAhrefs::rah_condition(
  column_name = "http_code",
  operator = "EQUALS",
```
value = "404")

# joining conditions into one condition set
cond_where <- RAhrefs::rah_condition_set(cond_1, cond_2)

# downloading
b <- RAhrefs::rah_broken_links(
  target = "ahrefs.com",
  limit  = 2,
  where  = cond_where,
  order_by = "domain_rating:desc")

## End(Not run)

rah_condition

Single condition creation for an Ahrefs API query

Description

This function creates an optional single condition for report querying. It can only be used in `where` and `having` parameters and should only be used inside `rah_condition_set` function.

Usage

`rah_condition(column_name, operator, value, is_date = FALSE)`

Arguments

- `column_name`: character string. Proper name of the column of the report to query from.
- `operator`: character string. See more in details.
- `value`: character string or numeric/integer. Contains the value for a condition.
- `is_date`: logical, defaults to FALSE. If provided value is a date character string, should be set to TRUE. Works only for dates in 'YYYY-MM-DD' format.

Details

This function should be ALWAYS used inside `rah_condition_set` function. Options include:

- "SUBDOMAIN" (string) the condition is satisfied if a domain in the `<column>` is a subdomain of the provided `<domain>`, for example: `rah_condition("url_to", "SUBDOMAIN", "dev")`
- "SUBSTRING" (string) the condition is satisfied if the provided `<value>` is a substring of the `<column>`, for example: `rah_condition("url_to", "SUBSTRING", "ample")`
- "WORD" (string) the condition is satisfied if the provided `<value>` appears as a separate word of the `<column>`, for example: `rah_condition("title", "WORD", "the")`
- "EQUALS", "UNEQUALS", "LESS_THAN", "LESS_OR_EQUAL", "GREATER_THAN", "GREATER_OR_EQUAL" (numeric/date) the condition is satisfied if a `<column>` is different to `<value>`, for example: `rah_condition("domain_rank", "GREATER_OR_EQUAL", 5)` OR for dates: `rah_condition("first_seen", "LESS_THAN", "2019-01-01", is_date = TRUE)`
**rah_condition_set**

**Value**

character string with formatted condition

**See Also**

Other Ahrefs conditions: **rah_condition_set**

**Examples**

```r
## Not run: rah_condition(column_name = "first_seen",
operator = "GREATER_THAN",
value = "2018-01-01",
is_date = TRUE)

## End(Not run)

## Not run: rah_condition(column_name = "links",
operator = "GREATER_THAN",
value = "10")

## End(Not run)
```

**Description**

Grouping multiple conditions for an Ahrefs API query

**Usage**

`rah_condition_set(...)`

**Arguments**

... multiple condition arguments created by **rah_condition** function

**Value**

character string of parameters for API

**See Also**

Other Ahrefs conditions: **rah_condition**
Examples

```r
## Not run:
first_condition <- RAhrefs::rah_condition(  
column_name = "first_seen",  
operator = "GREATER_THAN",  
value = "2018-01-01",  
is_date = TRUE)

second_condition <- RAhrefs::rah_condition(  
column_name = "links",  
operator = "GREATER_THAN",  
value = "10")

final_condition <- RAhrefs::rah_condition_set(  
  first_condition,  
  second_condition)

## End(Not run)
```

rah_domain_rating  Export the Domain Rating.

Description


Usage

```r
rah_domain_rating(target, token = Sys.getenv("AHREFS_AUTH_TOKEN"),  
  mode = "domain", metrics = NULL, limit = 1000, order_by = NULL,  
  where = NULL, having = NULL)
```

Arguments

target  character string. Aim of a request: a domain, a directory or a URL
token  character string. Authentication token. Should be available through environmental variables after authentication with function `rah_auth()`
mode  character string. Mode of operation: exact, domain, subdomains or prefix. See more in Details section
metrics  character vector of columns to select. See more in Details section
limit  integer. Number of results to return
order_by  character vector of columns to sort on. See more in Details section
where  character string - a condition created by `rah_condition_set()` function that generates proper "where" condition to satisfy. See more in Details section
having character string - a condition created by `rah_condition_set()` function that generates proper "having" condition to satisfy. See more in Details section

Details

1. available metrics - you can select which columns (metrics) you want to download and which one would be useful in filtering, BUT not all of them can always be used in "where" & "having" conditions:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Where</th>
<th>Having</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>domain_rating</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>Domain Rating of the target domain.</td>
</tr>
<tr>
<td>ahrefs_top</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>Ahrefs Rating of the target domain.</td>
</tr>
</tbody>
</table>

2. "mode" parameter can take 4 different values that will affect how the results will be grouped.

Example of URL directory with folder:

- **Example URL**: ahrefs.com/api/
  - **exact**: ahrefs.com/api/
  - **domain**: ahrefs.com/
  - **subdomains**: ahrefs.com/*
  - **prefix**: ahrefs.com/api/*

Example of URL directory with subdomain:

- **Example URL**: api2.ahrefs.com
  - **exact**: api2.ahrefs.com/
  - **domain**: api2.ahrefs.com/
  - **subdomains**: api2.ahrefs.com/*
  - **prefix**: api2.ahrefs.com/*

3. "order_by" parameter is a character string that forces sorting of the results. Structure:

- **Structure**: "column_name:asc|desc"
- **Single column example**: "first_seen:asc" - this sorts results by first_seen column in ascending order
- **Multi column example**: "last_seen:desc,first_seen:asc" - this sorts results by 1) last_seen column in descending order, and next by 2) first_seen column in ascending order

4. "where" & "having" are **EXPERIMENTAL** parameters of condition sets (character strings) that control filtering the results. To create arguments:

1. use `rah_condition()` function to create a single condition, for example: `cond_1 <- rah_condition(column = "links", operator = "GREATER_THAN", value = "10")`
2. use `rah_condition_set()` function to group single conditions into final condition string, for example: `fin_cond <- rah_condition_set(cond_1, cond_2)`
3. provide final condition to proper report function as a parameter, for example: `RAhrefs::rah_domain_rating(target = "ahrefs.com", token = "0123456789", mode = "domain", metrics = NULL, limit = 1000, where = fin_cond, order_by = "first_seen:asc")`
**Value**

data frame

**Source**

https://ahrefs.com/api/documentation

**See Also**

Other Ahrefs reports: `rah_ahrefs_rank`, `rah_anchors_refdomains`, `rah_anchors`, `rah_backlinks_new_lost_counters`, `rah_backlinks_new_lost`, `rah_backlinks_one_per_domain`, `rah_backlinks`, `rah_broken_backlinks`, `rah_broken_links`, `rah_linked_anchors`, `rah_linked_domains_by_type`, `rah_linked_domains`, `rah_metrics_extended`, `rah_metrics`, `rah_pages_extended`, `rah_pages_info`, `rah_pages`, `rah_refdomains_by_type`, `rah_refdomains_new_lost_counters`, `rah_refdomains_new_lost`, `rah_refdomains`, `rah_refips`, `rah_subscription_info`

**Examples**

```r
## Not run:
# creating single conditions for 'where' parameter
cond_1 <- RAhrefs::rah_condition(  
column_name = "domain_rating",  
operator = "GREATER_OR_EQUAL",  
value = "10")

cond_2 <- RAhrefs::rah_condition(  
column_name = "ahrefs_rank",  
operator = "GREATER_OR_EQUAL",  
value = "10")

# joining conditions into one condition set
cond_having <- RAhrefs::rah_condition_set(cond_1, cond_2)

# downloading
b <- RAhrefs::rah_anchors(  
target = "ahrefs.com",  
limit = 2,  
having = cond_having,  
order_by = "ahrefs_rank:desc")

## End(Not run)
```
Description

This is a helper function and it should not be used in most cases. Use `rah_<report_name>()` functions instead as they are specific wrappers that provide full documentation needed for each report.

Usage

```r
downloader(target, report, token = Sys.getenv("AHREFS_AUTH_TOKEN"),
            mode = "domain", metrics = NULL, limit = 1000, order_by = NULL,
            where = NULL, having = NULL)
```

Arguments

- **target**: character string. Aim of a request: a domain, a directory or a URL
- **report**: character string. Name of the table (report) to select data from
- **token**: character string. Authentication token. Should be available through environmental variables after authentication with function `rah_auth()`
- **mode**: character string. Mode of operation: exact, domain, subdomains or prefix. See more in Details section
- **metrics**: character vector of columns to select
- **limit**: integer. Number of results to return
- **order_by**: character vector of columns to sort on. See more in Details section
- **where**: character string - a condition created by `rah_condition_set` function that generates proper "Where" condition to satisfy. See more in Details section
- **having**: character string - a condition created by `rah_condition_set` function that generates proper "Having" condition to satisfy. See more in Details section

Details

1. "mode" parameter can take 4 different values that will affect how the results will be grouped.

Example of URL directory with folder:

- **Example URL**: ahrefs.com/api/
- **exact**: ahrefs.com/api/
- **domain**: ahrefs.com/*
- **subdomains**: *ahrefs.com/*
- **prefix**: ahrefs.com/api/*

Example of URL directory with subdomain:

- **Example URL**: apiv2.ahrefs.com
- **exact**: apiv2.ahrefs.com/
- **domain**: apiv2.ahrefs.com/*
- **subdomains**: *apiv2.ahrefs.com/*
2. "order_by" parameter is a character string that forces sorting of the results. Structure:
   - Structure: "column_name:asc|desc"
   - Single column example: "first_seen:asc" ~ this sorts results by first_seen column in ascending order
   - Multi column example: "last_seen:desc,first_seen:asc" ~ this sorts results by 1) last_seen column in descending order, and next by 2) first_seen column in ascending order

3. "where" & "having" are EXPERIMENTAL parameters of condition sets (character strings) that control filtering the results. To create arguments:
   1. use `rah_condition()` function to create a single condition, for example: `cond_1 <- rah_condition(column_name = "links", operator = "GREATER_THAN", value = "10")`
   2. use `rah_condition_set()` function to group single conditions into final condition string, for example: `fin_cond <- rah_condition_set(cond_1, cond_2)`
   3. provide final condition to proper report function as a parameter, for example: `RAhrefs::rah_downloader(target = "ahrefs.com", token = "0123456789", mode = "domain", metrics = NULL, limit = 1000, where = fin_cond, order_by = "first_seen:asc")`

Value

list or nested list object

Source

https://ahrefs.com/api/documentation

Examples

```r
# do not use this function - instead use its wrappers (rah_<report_name>())
# that have full documentation
## Not run: RAhrefs::rah_downloader(
target = "ahrefs.com",
report = "anchors",
token = "0123456789",
mode = "domain",
metrics = NULL,
limit = 1000,
where = rah_condition_set(
  rah_condition(column_name = "links",
                operator = "GREATER_THAN",
                value = "10"),
  rah_condition(column_name = "links",
                operator = "LESS_THAN",
                value = "20")),
order_by = "first_seen:asc")
## End(Not run)
```
**rah_linked_anchors**  
*Export the anchor text and the number of outgoing links that have it.*

---

**Description**

Export the anchor text and the number of outgoing links that have it.

**Usage**

```r
rah_linked_anchors(target, token = Sys.getenv("AHREFS_AUTH_TOKEN"),
                   mode = "domain", metrics = NULL, limit = 1000, order_by = NULL,
                   where = NULL, having = NULL)
```

**Arguments**

- `target` character string. Aim of a request: a domain, a directory or a URL
- `token` character string. Authentication token. Should be available through environmental variables after authentication with function `rah_auth()`
- `mode` character string. Mode of operation: exact, domain, subdomains or prefix. See more in Details section
- `metrics` character vector of columns to select. See more in Details section
- `limit` integer. Number of results to return
- `order_by` character vector of columns to sort on. See more in Details section
- `where` character string - a condition created by `rah_condition_set()` function that generates proper "where" condition to satisfy. See more in Details section
- `having` character string - a condition created by `rah_condition_set()` function that generates proper "having" condition to satisfy. See more in Details section

**Details**

1. **available metrics** - you can select which columns (metrics) you want to download and which one would be useful in filtering, BUT **not all of them can always be used** in "where" & "having" conditions:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Where</th>
<th>Having</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>anchor</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Anchor text used in at least one outgoing link from the target domain.</td>
</tr>
<tr>
<td>links_internal</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Number of internal outgoing links found that are using the anchor text.</td>
</tr>
<tr>
<td>links_external</td>
<td>int</td>
<td>+</td>
<td></td>
<td>Number of external outgoing links found that are using the anchor text.</td>
</tr>
<tr>
<td>url_from</td>
<td>string</td>
<td>+</td>
<td>-</td>
<td>URL of the page where the outgoing link is found.</td>
</tr>
<tr>
<td>url_to</td>
<td>string</td>
<td>+</td>
<td>-</td>
<td>URL of the page the outgoing link is pointing to.</td>
</tr>
<tr>
<td>ahrefs_rank</td>
<td>int</td>
<td>+</td>
<td>-</td>
<td>URL Rating of the referring page.</td>
</tr>
<tr>
<td>domain_rating</td>
<td>int</td>
<td>+</td>
<td>-</td>
<td>Domain Rating of the referring domain.</td>
</tr>
<tr>
<td>ip</td>
<td>string</td>
<td>+</td>
<td>-</td>
<td>IP address of the page.</td>
</tr>
<tr>
<td>page_size</td>
<td>int</td>
<td>+</td>
<td>-</td>
<td>Size of the referring page, in bytes.</td>
</tr>
</tbody>
</table>
encoding | string | + | - | Character encoding of the referring page, for example "utf8" or "iso-8859-1" (Latin-1).
title | string | + | - | Title of the referring page.
first_seen | date | + | - | Least recent date when the Ahrefs crawler was able to visit the backlink.
last_visited | date | + | - | Most recent date when the Ahrefs crawler was able to visit the backlink.
prev_visited | date | + | - | Second to the most recent date when the Ahrefs crawler was able to visit the backlink.
original | boolean | + | - | Indicates whether the backlink was present on the referring page when the Ahrefs crawler first visited it.
link_type | string | + | - | Either "href", "redirect", "frame", "form", "canonical", "rss", or "alternate".
redirect | int | + | - | For redirected links, the Redirect Code (3XX), zero otherwise.
nofollow | boolean | + | - | Indicates whether the backlink is NoFollow.
alt | string | + | - | Alternative text of the image backlink, if exists.
text_pre | string | + | - | Snippet before the anchor text.
text_post | string | + | - | Snippet after the anchor text.

2. "mode" parameter can take 4 different values that will affect how the results will be grouped.

Example of URL directory with folder:

- Example URL: ahrefs.com/api/
- exact: ahrefs.com/api/
- domain: ahrefs.com/*
- subdomains: *ahrefs.com/*
- prefix: ahrefs.com/api/*

Example of URL directory with subdomain:

- Example URL: apiv2.ahrefs.com
- exact: apiv2.ahrefs.com/
- domain: apiv2.ahrefs.com/*
- subdomains: *apiv2.ahrefs.com/*
- prefix: apiv2.ahrefs.com/*

3. "order_by" parameter is a character string that forces sorting of the results. Structure:

- Structure: "column_name:asc|desc"
- Single column example: "first_seen:asc" - this sorts results by first_seen column in ascending order
- Multi column example: "last_seen:desc.first_seen:asc" - this sorts results by 1) last_seen column in descending order, and next by 2) first_seen column in ascending order

4. "where" & "having" are EXPERIMENTAL parameters of condition sets (character strings) that control filtering the results. To create arguments:

1. use rah_condition() function to create a single condition, for example:
   ```r
   cond_1 <- rah_condition(column = "links", operator = "GREATER_THAN", value = "10")
   ```
2. use rah_condition_set() function to group single conditions into final condition string, for example:
   ```r
   fin_cond <- rah_condition_set(cond_1, cond_2)
   ```
3. provide final condition to proper report function as a parameter, for example:
   ```r
   RAhrefs::rah_linked_anchors(target = "ahrefs.com", token = "0123456789", mode = "domain", metrics = NULL, limit = 1000, where = fin_cond, order_by = "first_seen:asc")
   ```
rah_linked_domains

Value

data frame

Source

https://ahrefs.com/api/documentation

See Also

Other Ahrefs reports: rah_ahrefs_rank, rah_anchors_refdomains, rah_anchors, rah_backlinks_new_lost_counters, rah_backlinks_new_lost, rah_backlinks_one_per_domain, rah_backlinks, rah_broken_backlinks, rah_broken_links, rah_domain_rating, rah_linked_domains_by_type, rah_linked_domains, rah_metrics_extended, rah_metrics, rah_pages_extended, rah_pages_info, rah_pages, rah_refdomains_by_type, rah_refdomains_new_lost_counters, rah_refdomains_new_lost, rah_refdomains, rah_refips, rah_subscription_info

Examples

```r
## Not run:
# creating single conditions for 'where' parameter
cond_1 <- RAhrefs::rah_condition(
    column_name = "domain_rating",
    operator    = "GREATER_OR_EQUAL",
    value       = "10")

cond_2 <- RAhrefs::rah_condition(
    column_name = "ahrefs_rank",
    operator    = "GREATER_OR_EQUAL",
    value       = "10")

# joining conditions into one condition set
cond_where <- RAhrefs::rah_condition_set(cond_1, cond_2)

# downloading
b <- RAhrefs::rah_linked_anchors(
    target    = "ahrefs.com",
    limit      = 2,
    where      = cond_where,
    order_by   = "ahrefs_rank:desc")

## End(Not run)
```

rah_linked_domains  Export the external domains that the target has links to.

Description

Export the external domains that the target has links to.
Usage

rah_linked_domains(target, token = Sys.getenv("AHREFS_AUTH_TOKEN"),
    mode = "domain", metrics = NULL, limit = 1000, order_by = NULL,
    where = NULL, having = NULL)

Arguments

target character string. Aim of a request: a domain, a directory or a URL.
token character string. Authentication token. Should be available through enviromen-
tal variables after authentication with function rah_auth().
mode character string. Mode of operation: exact, domain, subdomains or prefix. See
more in Details section
metrics character vector of columns to select. See more in Details section
limit integer. Number of results to return
order_by character vector of columns to sort on. See more in Details section
where character string - a condition created by rah_condition_set() function
that generates proper "where" condition to satisfy. See more in Details section
having character string - a condition created by rah_condition_set() function
that generates proper "having" condition to satisfy. See more in Details sec-
tion

Details

1. available metrics - you can select which columns (metrics) you want to download and which one
would be useful in filtering, BUT not all of them can always be used in "where" & "having"

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Where</th>
<th>Having</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>domain_from</td>
<td>string</td>
<td>-</td>
<td>+</td>
<td>Target of the request.</td>
</tr>
<tr>
<td>domain_to</td>
<td>string</td>
<td>-</td>
<td>+</td>
<td>External domain that has links from the target.</td>
</tr>
<tr>
<td>links</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>Number of external links from the target that link to the external domain.</td>
</tr>
<tr>
<td>unique_pages</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>Number of unique pages from the target that link to the external domain.</td>
</tr>
<tr>
<td>domain_to_rating</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>Domain Rating of the external domain.</td>
</tr>
<tr>
<td>domain_to_ahrefs_top</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>Ahrefs rank of the external domain.</td>
</tr>
</tbody>
</table>

2. "mode" parameter can take 4 different values that will affect how the results will be grouped.

Example of URL directory with folder:
- Example URL: ahrefs.com/api/
- exact: ahrefs.com/api/
- domain: ahrefs.com/*
- subdomains: *ahrefs.com/*
- prefix: ahrefs.com/api/*

Example of URL directory with subdomain:
**Example URL**: apiv2.ahrefs.com

**exact**: apiv2.ahrefs.com/

**domain**: apiv2.ahrefs.com/*

**subdomains**: *apiv2.ahrefs.com/*

**prefix**: apiv2.ahrefs.com/*

3. "order_by" parameter is a character string that forces sorting of the results. Structure:

- **Structure**: "column_name:asc|desc"
- **Single column example**: "first_seen:asc" - this sorts results by first_seen column in ascending order
- **Multi column example**: "last_seen:desc,first_seen:asc" - this sorts results by 1) last_seen column in descending order, and next by 2) first_seen column in ascending order

4. "where" & "having" are EXPERIMENTAL parameters of condition sets (character strings) that control filtering the results. To create arguments:

1. use rah_condition() function to create a single condition, for example: cond_1 <- rah_condition(column = "links", operator = "GREATER_THAN", value = "10")
2. use rah_condition_set() function to group single conditions into final condition string, for example: fin_cond <- rah_condition_set(cond_1, cond_2)
3. provide final condition to proper report function as a parameter, for example: RAhrefs::rah_linked_domains(target = "ahrefs.com", token = "0123456789", mode = "domain", metrics = NULL, limit = 1000, where = fin_cond, order_by = "first_seen:asc")

**Value**

data frame

**Source**

https://ahrefs.com/api/documentation

https://ahrefs.com/api/documentation

**See Also**

Other Ahrefs reports: rah_ahrefs_rank, rah_anchors_refdomains, rah_anchors, rah_backlinks_new_lost_counters, rah_backlinks_new_lost, rah_backlinks_one_per_domain, rah_backlinks, rah_broken_backlinks, rah_broken_links, rah_domain_rating, rah_linked_anchors, rah_linked_domains_by_type, rah_metrics_extended, rah_metrics, rah_pages_extended, rah_pages_info, rah_pages, rah_refdomains_by_type, rah_refdomains_new_lost_counters, rah_refdomains_new_lost, rah_refdomains, rah_refips, rah_subscription_info
Examples

```r
# creating single conditions for 'having' parameter
cond_1 <- RAhrefs::rah_condition(
  column_name = "domain_to_rating",
  operator    = "GREATER_OR_EQUAL",
  value       = "10")

cond_2 <- RAhrefs::rah_condition(
  column_name = "unique_pages",
  operator    = "GREATER_OR_EQUAL",
  value       = "3")

# joining conditions into one condition set
cond_having <- RAhrefs::rah_condition_set(cond_1, cond_2)

# downloading
b <- RAhrefs::rah_linked_domains(
  target = "ahrefs.com",
  limit  = 2,
  having = cond_having,
  order_by = "ahrefs_rank:desc")
```

## End(Not run)

### `rah_linked_domains_by_type`

Export the external domains that the target has links to.

**Description**

Export the external domains that the target has links to.

**Usage**

`rah_linked_domains_by_type(target,  
  token = Sys.getenv("AHREFS_AUTH_TOKEN"), mode = "domain",  
  metrics = NULL, limit = 1000, order_by = NULL, where = NULL,  
  having = NULL)`

**Arguments**

- **target**: character string. Aim of a request: a domain, a directory or a URL
- **token**: character string. Authentication token. Should be available through environmental variables after authentication with function `rah_auth()`
- **mode**: character string. Mode of operation: exact, domain, subdomains or prefix. See more in Details section
- **metrics**: character vector of columns to select. See more in Details section
**rah_linked_domains_by_type**

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Where</th>
<th>Having</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>domain_from</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Target of the request.</td>
</tr>
<tr>
<td>domain_to</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Root of external domain that has links from the target.</td>
</tr>
<tr>
<td>links</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>Number of external links from the target that link to the external domain.</td>
</tr>
<tr>
<td>all</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>Number of external links from the target that link to the external domain.</td>
</tr>
<tr>
<td>text</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>Number of text external links from the target.</td>
</tr>
<tr>
<td>image</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>Number of image external links from the target.</td>
</tr>
<tr>
<td>nofollow</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>Number of NoFollow external links from the target.</td>
</tr>
<tr>
<td>dofollow</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>Number of DoFollow external links from the target.</td>
</tr>
<tr>
<td>redirect</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>Number of redirection external links from the target.</td>
</tr>
<tr>
<td>canonical</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>Number of canonical external links from the target.</td>
</tr>
<tr>
<td>gov</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>Number of external links from the target to governmental domain.</td>
</tr>
<tr>
<td>edu</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>Number of external links from the target to educational domain.</td>
</tr>
<tr>
<td>rss</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>Number of RSS external links from the target.</td>
</tr>
<tr>
<td>alternate</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>Number of alternate external links from the target.</td>
</tr>
<tr>
<td>first_seen</td>
<td>date</td>
<td>+</td>
<td>+</td>
<td>Least recent date when the Ahrefs crawler was able to see the link on the target</td>
</tr>
<tr>
<td>last_visited</td>
<td>date</td>
<td>+</td>
<td>+</td>
<td>Most recent date when the Ahrefs crawler was able to see the link on the target</td>
</tr>
<tr>
<td>domain_to_rating</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>Domain Rating of the external domain.</td>
</tr>
<tr>
<td>domain_to_ahrefs_top</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>Ahrefs rank of the external domain.</td>
</tr>
</tbody>
</table>

**Details**

1. **available metrics** - you can select which columns (metrics) you want to download and which one would be useful in filtering, BUT not all of them can always be used in "where" & "having" conditions:

2. **"mode" parameter** can take 4 different values that will affect how the results will be grouped.

Example of URL directory with folder:

- **Example URL**: ahrefs.com/api/
- **exact**: ahrefs.com/api/
- **domain**: ahrefs.com/*
- **subdomains**: *ahrefs.com/*
- **prefix**: ahrefs.com/api/*

Example of URL directory with subdomain:

- **Example URL**: apiv2.ahrefs.com
- **exact**: apiv2.ahrefs.com/
• **domain**: apiv2.ahrefs.com/*
• **subdomains**: *apiv2.ahrefs.com/*
• **prefix**: apiv2.ahrefs.com/*

3. "**order_by**" parameter is a character string that forces sorting of the results. Structure:

   • **Structure**: "column_name:asc|desc"
   • **Single column example**: "first_seen:asc" ~ this sorts results by first_seen column in ascending order
   • **Multi column example**: "last_seen:desc,first_seen:asc" ~ this sorts results by 1) last_seen column in descending order, and next by 2) first_seen column in ascending order

4. "**where**" & "**having**" are **EXPERIMENTAL** parameters of condition sets (character strings) that control filtering the results. To create arguments:

1. use **rah_condition()** function to create a single condition, for example: `cond_1 <- RAhrefs::rah_condition(column = "all", operator = "GREATER_OR_EQUAL", value = "10")`
2. use **rah_condition_set()** function to group single conditions into final condition string, for example: `fin_cond <- RAhrefs::rah_condition_set(cond_1, cond_2)`
3. provide final condition to proper report function as a parameter, for example: `RAhrefs::rah_linked_domains_by_type(target = "ahrefs.com", token = "0123456789", mode = "domain", metrics = NULL, limit = 1000, where = fin_cond, order_by = "first_seen:asc")`

**Value**

data frame

**Source**

https://ahrefs.com/api/documentation

**See Also**

Other Ahrefs reports: rah_ahrefs_rank, rah_anchors_refdomains, rah_anchors, rah_backlinks_new_lost_counts, rah_backlinks_new_lost, rah_backlinks_one_per_domain, rah_backlinks, rah_broken_backlinks, rah_broken_links, rah_domain_rating, rah_linked_anchors, rah_linked_domains, rah_metrics_extended, rah_metrics, rah_pages_extended, rah_pages_info, rah_pages, rah_refdomains_by_type, rah_refdomains_new_lost_counts, rah_refdomains_new_lost, rah_refdomains, rah_refips, rah_subscription_info

**Examples**

```r
# Not run:
# creating single conditions for 'having' parameter
cond_1 <- RAhrefs::rah_condition(
  column_name = "all",
  operator = "GREATER_OR_EQUAL",
  value = "10")
```
cond_2 <- RAhrefs::rah_condition(
  column_name = "unique_pages",
  operator = "GREATER_OR_EQUAL",
  value = "6")

# joining conditions into one condition set
cond_having <- RAhrefs::rah_condition_set(cond_1, cond_2)

# downloading
b <- RAhrefs::rah_linked_domains_by_type(
  target = "ahrefs.com",
  limit = 2,
  having = cond_having,
  order_by = "ahrefs_rank:desc")

## End(Not run)

### Description
Export metrics about the target, such as total number of backlinks, referring pages, etc., that are similar to the Site Explorer Overview page with the addition of stats for total number of HTML pages, internal and external links.

### Usage

```r
rah_metrics(target, token = Sys.getenv("AHREFS_AUTH_TOKEN"),
  mode = "domain", metrics = NULL, limit = 1000, order_by = NULL,
  where = NULL, having = NULL)
```

### Arguments

- **target**: character string. Aim of a request: a domain, a directory or a URL
- **token**: character string. Authentication token. Should be available through environmental variables after authentication with function `rah_auth()`
- **mode**: character string. Mode of operation: exact, domain, subdomains or prefix. See more in Details section
- **metrics**: character vector of columns to select. See more in Details section
- **limit**: integer. Number of results to return
- **order_by**: character vector of columns to sort on. See more in Details section
where character string - a condition created by `rah_condition_set()` function that generates proper "where" condition to satisfy. See more in Details section

having character string - a condition created by `rah_condition_set()` function that generates proper "having" condition to satisfy. See more in Details section

**Details**

1. **available metrics** - you can select which columns (metrics) you want to download and which one would be useful in filtering, **BUT not all of them can always be used** in "where" & "having" conditions:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Where</th>
<th>Having</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>backlinks</td>
<td>int</td>
<td>-</td>
<td>-</td>
<td>Number of external backlinks found on the referring pages that link to the target.</td>
</tr>
<tr>
<td>refpages</td>
<td>int</td>
<td>-</td>
<td>-</td>
<td>Number of external web pages containing at least one backlink that links to the target.</td>
</tr>
<tr>
<td>pages</td>
<td>int</td>
<td>-</td>
<td>-</td>
<td>Number of unique pages visited by the Ahrefs crawler on the target.</td>
</tr>
<tr>
<td>text</td>
<td>int</td>
<td>-</td>
<td>-</td>
<td>Number of backlinks that use anchor texts.</td>
</tr>
<tr>
<td>image</td>
<td>int</td>
<td>-</td>
<td>-</td>
<td>Number of backlinks that use image as an anchor.</td>
</tr>
<tr>
<td>nofollow</td>
<td>int</td>
<td>-</td>
<td>-</td>
<td>Number of NoFollow backlinks that link to the target.</td>
</tr>
<tr>
<td>dofollow</td>
<td>int</td>
<td>-</td>
<td>-</td>
<td>Number of DoFollow backlinks that link to the target.</td>
</tr>
<tr>
<td>redirect</td>
<td>int</td>
<td>-</td>
<td>-</td>
<td>Number of redirects found that forward to the target.</td>
</tr>
<tr>
<td>canonical</td>
<td>int</td>
<td>-</td>
<td>-</td>
<td>Number of canonical backlinks that link to the target.</td>
</tr>
<tr>
<td>gov</td>
<td>int</td>
<td>-</td>
<td>-</td>
<td>Number of backlinks of all types (including images and NoFollow) found on web pages on governmental domains that link to the target.</td>
</tr>
<tr>
<td>edu</td>
<td>int</td>
<td>-</td>
<td>-</td>
<td>Number of backlinks of all types (including images and NoFollow) found on web pages on educational domains that link to the target.</td>
</tr>
<tr>
<td>html_pages</td>
<td>int</td>
<td>-</td>
<td>-</td>
<td>Number of HTML pages the target link has.</td>
</tr>
<tr>
<td>links_internal</td>
<td>int</td>
<td>-</td>
<td>-</td>
<td>Number of internal links found in the target.</td>
</tr>
<tr>
<td>links_external</td>
<td>int</td>
<td>-</td>
<td>-</td>
<td>Number of external links found in the target.</td>
</tr>
</tbody>
</table>

2. **"mode"** parameter can take 4 different values that will affect how the results will be grouped.

Example of URL directory with folder:

- **Example URL**: ahrefs.com/api/
- **exact**: ahrefs.com/api/
- **domain**: ahrefs.com/*
- **subdomains**: *ahrefs.com/*
- **prefix**: ahrefs.com/api/*

Example of URL directory with subdomain:

- **Example URL**: apiv2.ahrefs.com
- **exact**: apiv2.ahrefs.com/
- **domain**: apiv2.ahrefs.com/*
- **subdomains**: *apiv2.ahrefs.com/*
- **prefix**: apiv2.ahrefs.com/*

3. **"order_by"** parameter is a character string that forces sorting of the results. Structure:
• **Structure:** "column_name:asc|desc"

• **Single column example:** "first_seen:asc" ~ this sorts results by `first_seen` column in ascending order

• **Multi column example:** "last_seen:desc,first_seen:asc" ~ this sorts results by 1) `last_seen` column in descending order, and next by 2) `first_seen` column in ascending order

4. "where" & "having" are **EXPERIMENTAL** parameters of condition sets (character strings) that control filtering the results. To create arguments:

1. use `rah_condition()` function to create a single condition, for example: `cond_1 <- rah_condition(column = "links", operator = "GREATER_THAN", value = "10")`

2. use `rah_condition_set()` function to group single conditions into final condition string, for example: `fin_cond <- rah_condition_set(cond_1, cond_2)`

3. provide final condition to proper report function as a parameter, for example: `RAhrefs::rah_metrics(target = "ahrefs.com", token = "0123456789", mode = "domain", metrics = NULL, limit = 1000, where = fin_cond, order_by = "first_seen:asc")`

**Value**

data frame

**Source**

https://ahrefs.com/api/documentation

**See Also**

Other Ahrefs reports: `rah_ahrefs_rank`, `rah_anchors_refdomains`, `rah_anchors`, `rah_backlinks_new_lost_counters`, `rah_backlinks_new_lost`, `rah_backlinks_one_per_domain`, `rah_backlinks`, `rah_broken_backlinks`, `rah_broken_links`, `rah_domain_rating`, `rah_linked_anchors`, `rah_linked_domains_by_type`, `rah_linked_domains`, `rah_metrics_extended`, `rah_pages_extended`, `rah_pages_info`, `rah_pages`, `rah_refdomains_by_type`, `rah_refdomains_new_lost_counters`, `rah_refdomains_new_lost`, `rah_refdomains`, `rah_refips`, `rah_subscription_info`

**Examples**

```r
# Not run:
# downloading
b <- RAhrefs::rah_metrics(
  target = "ahrefs.com",
  limit = 2,
  order_by = "backlinks:desc")

# End(Not run)
```
Export additional metrics about the target, such as total number of referring domains, referring class C networks and referring IP addresses.

**Description**

Export additional metrics about the target, such as total number of referring domains, referring class C networks and referring IP addresses.

**Usage**

```r
rah_metrics_extended(target, token = Sys.getenv("AHREFS_AUTH_TOKEN"),
                     mode = "domain", metrics = NULL, limit = 1000, order_by = NULL,
                     where = NULL, having = NULL)
```

**Arguments**

- **target** character string. Aim of a request: a domain, a directory or a URL.
- **token** character string. Authentication token. Should be available through environmental variables after authentication with function `rah_auth()`.
- **mode** character string. Mode of operation: exact, domain, subdomains or prefix. See more in Details section.
- **metrics** character vector of columns to select. See more in Details section.
- **limit** integer. Number of results to return.
- **order_by** character vector of columns to sort on. See more in Details section.
- **where** character string - a condition created by `rah_condition_set()` function that generates proper "where" condition to satisfy. See more in Details section.
- **having** character string - a condition created by `rah_condition_set()` function that generates proper "having" condition to satisfy. See more in Details section.

**Details**

1. **Available metrics** - you can select which columns (metrics) you want to download and which one would be useful in filtering, **BUT not all of them can always be used** in "where" & "having" conditions:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Where</th>
<th>Having</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>backlinks</td>
<td>int</td>
<td>-</td>
<td>-</td>
<td>Number of external backlinks found on the referring pages that link to the target.</td>
</tr>
<tr>
<td>refpages</td>
<td>int</td>
<td>-</td>
<td>-</td>
<td>Number of external web pages containing at least one backlink that links to the target.</td>
</tr>
<tr>
<td>pages</td>
<td>int</td>
<td>-</td>
<td>-</td>
<td>Number of unique pages visited by the Ahrefs crawler on the target.</td>
</tr>
<tr>
<td>valid_pages</td>
<td>int</td>
<td>-</td>
<td>-</td>
<td>Number of unique pages with non 5xx HTTP code, visited by the Ahrefs crawler on the target.</td>
</tr>
<tr>
<td>text</td>
<td>int</td>
<td>-</td>
<td>-</td>
<td>Number of backlinks that use anchor texts.</td>
</tr>
</tbody>
</table>
image int - - Number of backlinks that use image as an anchor.
nofollow int - - Number of NoFollow backlinks that link to the target.
dofollow int - - Number of DoFollow backlinks that link to the target.
redirect int - - Number of redirects found that forward to the target.
canonical int - - Number of canonical backlinks that link to the target.
alternate int - - Number of alternate backlinks that link to the target.
gov int - - Number of backlinks of all types (including images and NoFollow) found on governmental domains that link to the target.
edu int - - Number of backlinks of all types (including images and NoFollow) found on educational domains that link to the target.
rss int - - Number of RSS external links from the target.
html_pages int - - Number of HTML pages the target link has.
links_internal int - - Number of internal links found in the target.
links_external int - - Number of external links found in the target.
refdomains int - - Number of domains containing at least one backlink that links to the target.
refclass_c int - - Number of referring class C networks that link to the target.
refips int - - Number of distinct IP addresses under a single network that link to the target.
linked_root_domains int - - Number of internal or external domains that are linked from the target.

2. "mode" parameter can take 4 different values that will affect how the results will be grouped.

Example of URL directory with folder:

- Example URL: ahrefs.com/api/
  - exact: ahrefs.com/api/
  - domain: ahrefs.com/*
  - subdomains: *ahrefs.com/*
  - prefix: ahrefs.com/*

Example of URL directory with subdomain:

- Example URL: apiv2.ahrefs.com
  - exact: apiv2.ahrefs.com/
  - domain: apiv2.ahrefs.com/*
  - subdomains: *apiv2.ahrefs.com/*
  - prefix: apiv2.ahrefs.com/*

3. "order_by" parameter is a character string that forces sorting of the results. Structure:

- Structure: "column_name:asc|desc"
- Single column example: "first_seen:asc" - this sorts results by first_seen column in ascending order
- Multi column example: "last_seen:desc,first_seen:asc" - this sorts results by 1) last_seen column in descending order, and next by 2) first_seen column in ascending order

4. "where" & "having" are EXPERIMENTAL parameters of condition sets (character strings) that control filtering the results. To create arguments:

1. use rah_condition() function to create a single condition, for example: cond_1 <- rah_condition(column = "links", operator = "GREATER_THAN", value = "10")
2. use `rah_condition_set()` function to group single conditions into final condition string, for example: `fin_cond <- rah_condition_set(cond_1, cond_2)`

3. provide final condition to proper report function as a parameter, for example: 
   ```r
   RAhrefs::rah_metrics_extended(target = "ahrefs.com", token = "0123456789", mode = "domain", metrics = NULL, limit = 1000, where = fin_cond, order_by = "first_seen:asc")
   ```

Value
data frame

Source
https://ahrefs.com/api/documentation

See Also
Other Ahrefs reports: 
- `rah_ahrefs_rank`
- `rah_anchors_refdomains`
- `rah_anchors`
- `rah_backlinks_new_lost_counters`
- `rah_backlinks_new_lost`
- `rah_backlinks_one_per_domain`
- `rah_backlinks`
- `rah_broken_backlinks`
- `rah_broken_links`
- `rah_domain_rating`
- `rah_linked_anchors`
- `rah_linked_domains_by_type`
- `rah_linked_domains`
- `rah_metrics`
- `rah_pages_extended`
- `rah_pages_info`
- `rah_pages`
- `rah_refdomains_by_type`
- `rah_refdomains_new_lost_counters`
- `rah_refdomains_new_lost`
- `rah_refdomains`
- `rah_refips`
- `rah_subscription_info`

Examples
```r
## Not run:
# downloading
b <- RAhrefs::rah_metrics_extended(
  target = "ahrefs.com",
  limit = 2,
  order_by = "backlinks:desc")
## End(Not run)
```
Arguments

- **target**: character string. Aim of a request: a domain, a directory or a URL.
- **token**: character string. Authentication token. Should be available through environmental variables after authentication with function `rah_auth()`.
- **mode**: character string. Mode of operation: exact, domain, subdomains or prefix. See more in Details section.
- **metrics**: character vector of columns to select. See more in Details section.
- **limit**: integer. Number of results to return.
- **order_by**: character vector of columns to sort on. See more in Details section.
- **where**: character string - a condition created by `rah_condition_set()` function that generates proper "where" condition to satisfy. See more in Details section.
- **having**: character string - a condition created by `rah_condition_set()` function that generates proper "having" condition to satisfy. See more in Details section.

Details

1. **available metrics** - you can select which columns (metrics) you want to download and which one would be useful in filtering, **BUT not all of them can always be used** in "where" & "having" conditions:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Where</th>
<th>Having</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>URL of the crawled page.</td>
</tr>
<tr>
<td>ahrefs_rank</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>URL Rating of the page.</td>
</tr>
<tr>
<td>first_seen</td>
<td>date</td>
<td>+</td>
<td>+</td>
<td>The date Ahrefs' bot first found a backlink to your target website or URL on</td>
</tr>
<tr>
<td>last_visited</td>
<td>date</td>
<td>+</td>
<td>+</td>
<td>Most recent date when the Ahrefs crawler was able to crawl the page.</td>
</tr>
<tr>
<td>http_code</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>HTTP code that was last returned for the page.</td>
</tr>
<tr>
<td>size</td>
<td>int</td>
<td>+</td>
<td></td>
<td>Size of the crawled page, in bytes.</td>
</tr>
<tr>
<td>links_internal</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Number of internal links found in the crawled page.</td>
</tr>
<tr>
<td>links_external</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Number of external links found in the crawled page.</td>
</tr>
<tr>
<td>encoding</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Character encoding of the page, for example &quot;utf8&quot; or &quot;iso-8859-1&quot; (Latin-1).</td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Title of the crawled page.</td>
</tr>
<tr>
<td>redirect_url</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>URL where the page redirects to.</td>
</tr>
<tr>
<td>content_encoding</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Type of encoding used to compress the page data, for example &quot;gzip&quot; or &quot;deflate&quot;.</td>
</tr>
</tbody>
</table>

2. **"mode"** parameter can take 4 different values that will affect how the results will be grouped. Example of URL directory with folder:

- **Example URL**: ahrefs.com/api/
- **exact**: ahrefs.com/api/
- **domain**: ahrefs.com/*
- **subdomains**: *ahrefs.com/*
- **prefix**: ahrefs.com/api/*

Example of URL directory with subdomain:
3. "order_by" parameter is a character string that forces sorting of the results. Structure:

- **Structure:** "column_name:asc|desc"
- **Single column example:** "first_seen:asc" ~ this sorts results by first_seen column in ascending order
- **Multi column example:** "last_seen:desc,first_seen:asc" ~ this sorts results by 1) last_seen column in descending order, and next by 2) first_seen column in ascending order

4. "where" & "having" are **EXPERIMENTAL** parameters of condition sets (character strings) that control filtering the results. To create arguments:

1. use `rah_condition()` function to create a single condition, for example: `cond_1 <- rah_condition(column = "links", operator = "GREATER_THAN", value = "10")`
2. use `rah_condition_set()` function to group single conditions into final condition string, for example: `fin_cond <- rah_condition_set(cond_1, cond_2)`
3. provide final condition to proper report function as a parameter, for example: `RAhrefs::rah_pages(target = "ahrefs.com", token = "0123456789", mode = "domain", metrics = NULL, limit = 1000, where = fin_cond, order_by = "first_seen:asc")`

Value
data frame

Source

https://ahrefs.com/api/documentation

See Also

**Other Ahrefs reports:** rah_ahrefs_rank, rah_anchors_refdomains, rah_anchors, rah_backlinks_new_lost_counters, rah_backlinks_new_lost, rah_backlinks_one_per_domain, rah_backlinks, rah_broken_backlinks, rah_broken_links, rah_domain_rating, rah_linked_anchors, rah_linked_domains_by_type, rah_linkedDomains, rah_metrics_extended, rah_metrics, rah_pages_extended, rah_pages_info, rah_refdomains_by_type, rah_refdomains_new_lost_counters, rah_refdomains_new_lost, rah_refdomains, rah_refips, rah_subscription_info
Examples

```r
## Not run:
# creating single conditions for 'where' parameter
cond_1 <- RAhrefs::rah_condition(
  column_name = "ahrefs_rank",
  operator = "GREATER_OR_EQUAL",
  value = "10")

cond_2 <- RAhrefs::rah_condition(
  column_name = "size",
  operator = "GREATER_THAN",
  value = "2048000")

# joining conditions into one condition set
cond_where <- RAhrefs::rah_condition_set(cond_1, cond_2)

# downloading
b <- RAhrefs::rah_pages(
  target = "ahrefs.com",
  limit = 2,
  where = cond_where,
  order_by = "ahrefs_rank:desc")
```

## End(Not run)

#### `rah_pages_extended`

Export additional metrics about the target, such as total number of referring domains, referring class C networks and referring IP addresses.

#### Description

Export additional metrics about the target, such as total number of referring domains, referring class C networks and referring IP addresses.

#### Usage

```r
rah_pages_extended(target, token = Sys.getenv("AHREFS_AUTH_TOKEN"),
                   mode = "domain", metrics = NULL, limit = 1000, order_by = NULL,
                   where = NULL, having = NULL)
```

#### Arguments

- **target** character string. Aim of a request: a domain, a directory or a URL
- **token** character string. Authentication token. Should be available through enviromental variables after authentication with function `rah_auth()`
- **mode** character string. Mode of operation: exact, domain, subdomains or prefix. See more in Details section
metrics character vector of columns to select. See more in Details section
limit integer. Number of results to return
order_by character vector of columns to sort on. See more in Details section
where character string - a condition created by `rah_condition_set()` function that generates proper "where" condition to satisfy. See more in Details section
having character string - a condition created by `rah_condition_set()` function that generates proper "having" condition to satisfy. See more in Details section

Details

1. available metrics - you can select which columns (metrics) you want to download and which one would be useful in filtering, BUT not all of them can always be used in "where" & "having" conditions:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Where</th>
<th>Having</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>URL of the crawled page.</td>
</tr>
<tr>
<td>ahrefs_rank</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>URL Rating of the page.</td>
</tr>
<tr>
<td>first_seen</td>
<td>date</td>
<td>+</td>
<td>+</td>
<td>Least recent date when the Ahrefs crawler was able to crawl the page.</td>
</tr>
<tr>
<td>last_visited</td>
<td>date</td>
<td>+</td>
<td>+</td>
<td>Most recent date when the Ahrefs crawler was able to crawl the page.</td>
</tr>
<tr>
<td>http_code</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>HTTP code that was last returned for the page.</td>
</tr>
<tr>
<td>size</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Size of the crawled page, in bytes.</td>
</tr>
<tr>
<td>links_internal</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Number of internal links found in the crawled page.</td>
</tr>
<tr>
<td>links_external</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Number of external links found in the crawled page.</td>
</tr>
<tr>
<td>encoding</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Character encoding of the page, for example &quot;utf8&quot; or &quot;iso-8859-1&quot; (Latin-1)</td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Title of the crawled page.</td>
</tr>
<tr>
<td>redirect_url</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>URL where the page redirects to.</td>
</tr>
<tr>
<td>content_encoding</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Type of encoding used to compress the page data, for example &quot;gzip&quot; or &quot;deflate&quot;</td>
</tr>
<tr>
<td>backlinks</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Number of external backlinks found on the referring pages that link to the crawled page.</td>
</tr>
<tr>
<td>dofollow</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Number of DoFollow backlinks that link to the crawled page.</td>
</tr>
<tr>
<td>nofollow</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Number of NoFollow backlinks that link to the crawled page.</td>
</tr>
<tr>
<td>redirects</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Number of pages found that redirect to the crawled page.</td>
</tr>
<tr>
<td>refdomains</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Number of domains containing at least one backlink that links to the crawled page.</td>
</tr>
<tr>
<td>refclass_c</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Number of referring class C networks that link to the crawled page.</td>
</tr>
<tr>
<td>refips</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Number of distinct IP addresses under a single network that link to the crawled page.</td>
</tr>
</tbody>
</table>

2. "mode" parameter can take 4 different values that will affect how the results will be grouped.

Example of URL directory with folder:

- **Example URL**: ahrefs.com/api/
- **exact**: ahrefs.com/api/
- **domain**: ahrefs.com/*
- **subdomains**: *ahrefs.com/*
- **prefix**: ahrefs.com/api/*

Example of URL directory with subdomain:
3. "order_by" parameter is a character string that forces sorting of the results. Structure:

- **Structure**: "column_name:asc|desc"
- **Single column example**: "first_seen:asc" ~ this sorts results by first_seen column in ascending order
- **Multi column example**: "last_seen:desc,first_seen:asc" ~ this sorts results by 1) last_seen column in descending order, and next by 2) first_seen column in ascending order

4. "where" & "having" are **EXPERIMENTAL** parameters of condition sets (character strings) that control filtering the results. To create arguments:

1. use `rah_condition()` function to create a single condition, for example: `cond_1 <- RAhrefs::rah_condition(column_name = "links", operator = "GREATER_THAN", value = "10")`
2. use `rah_condition_set()` function to group single conditions into final condition string, for example: `fin_cond <- RAhrefs::rah_condition_set(cond_1, cond_2)`
3. provide final condition to proper report function as a parameter, for example: `RAhrefs::rah_pages_extended(target = "ahrefs.com", token = "0123456789", mode = "domain", metrics = NULL, limit = 1000, where = fin_cond, order_by = "first_seen:asc")`

**Value**

- data frame

**Source**

https://ahrefs.com/api/documentation

**See Also**

- Other Ahrefs reports: `rah_ahrefs_rank`, `rah_anchors_refdomains`, `rah_anchors`, `rah_backlinks_new_lost_counters`, `rah_backlinks_new_lost`, `rah_backlinks_one_per_domain`, `rah_backlinks`, `rah_broken_backlinks`, `rah_broken_links`, `rah_domain_rating`, `rah_linked_anchors`, `rah_linked_domains_by_type`, `rah_linked_domains`, `rah_metrics_extended`, `rah_metrics`, `rah_pages_info`, `rah_pages`, `rah_refdomains_by_type`, `rah_refdomains_new_lost`, `rah_refdomains_new_lost`, `rah_refdomains`, `rah_refips`, `rah_subscription_info`

**Examples**

```r
# Not run:
# creating single conditions for 'where' parameter
cond_1 <- RAhrefs::rah_condition(
  column_name = "ahrefs_rank",
  operator = "GREATER_OR_EQUAL",
```
value = "10")

cond_2 <- RAhrefs::rah_condition(
    column_name = "size",
    operator = "GREATER_THAN",
    value = "2048000")

# joining conditions into one condition set
cond_where <- RAhrefs::rah_condition_set(cond_1, cond_2)

# downloading
b <- RAhrefs::rah_pages_extended(
    target = "ahrefs.com",
    limit = 2,
    where = cond_where,
    order_by = "ahrefs_rank:desc")

# End(Not run)

rah_pages_info

Export additional info about the target, such as IP address, canonical URL, social meta tags and social metrics.

Description
Export additional info about the target, such as IP address, canonical URL, social meta tags and social metrics.

Usage

rah_pages_info(target, token = Sys.getenv("AHREFS_AUTH_TOKEN"),
    mode = "domain", metrics = NULL, limit = 1000, order_by = NULL,
    where = NULL, having = NULL)

Arguments

target character string. Aim of a request: a domain, a directory or a URL
token character string. Authentication token. Should be available through enviromental variables after authentication with function rah_auth()
mode character string. Mode of operation: exact, domain, subdomains or prefix. See more in Details section
metrics character vector of columns to select. See more in Details section
limit integer. Number of results to return
order_by character vector of columns to sort on. See more in Details section
where character string - a condition created by rah_condition_set() function that generates proper "where" condition to satisfy. See more in Details section
having character string - a condition created by rah_condition_set() function that generates proper "having" condition to satisfy. See more in Details section
Details

1. **available metrics** - you can select which columns (metrics) you want to download and which one would be useful in filtering, **BUT not all of them can always be used** in "where" & "having" conditions:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Where</th>
<th>Having</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>URL of the crawled page.</td>
</tr>
<tr>
<td>ip</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>IP address of the server that returned the page.</td>
</tr>
<tr>
<td>size</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Size of the crawled page, in bytes.</td>
</tr>
<tr>
<td>links_internal</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Number of internal links found in the crawled page.</td>
</tr>
<tr>
<td>links_external</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Number of external links found in the crawled page.</td>
</tr>
<tr>
<td>encoding</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Character encoding of the page, for example &quot;utf8&quot; or &quot;iso-8859-1&quot; (Latin-1).</td>
</tr>
<tr>
<td>title</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Title of the crawled page.</td>
</tr>
<tr>
<td>redirect_url</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>URL where the page redirects to.</td>
</tr>
<tr>
<td>canonical_url</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Canonical URL of the page.</td>
</tr>
<tr>
<td>content_encoding</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Type of encoding used to compress the page data, for example &quot;gzip&quot;.</td>
</tr>
<tr>
<td>description</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Description of the crawled page.</td>
</tr>
<tr>
<td>meta_social</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Contents of meta tags for social sharing sites.</td>
</tr>
<tr>
<td>twitter</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Number of Twitter shares of the page.</td>
</tr>
<tr>
<td>pinterest</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Number of Pinterest shares of the page.</td>
</tr>
<tr>
<td>facebook_likes</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Number of Facebook likes of the page.</td>
</tr>
<tr>
<td>facebook_shares</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Number of Facebook shares of the page.</td>
</tr>
<tr>
<td>facebook_comments</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Number of Facebook comments of the page.</td>
</tr>
<tr>
<td>facebook_clicks</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Number of Facebook clicks of the page.</td>
</tr>
<tr>
<td>facebook_comments_box</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Number of Facebook box comments of the page.</td>
</tr>
<tr>
<td>facebook</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Total number of Facebook shares/likes of the page.</td>
</tr>
<tr>
<td>total_shares</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Total number of shares of the page across all social networks.</td>
</tr>
<tr>
<td>median_shares</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Median number of shares of the page across all social networks.</td>
</tr>
</tbody>
</table>

2. **"mode"** parameter can take 4 different values that will affect how the results will be grouped.

Example of URL directory with folder:

- **Example URL**: ahrefs.com/api/
  - **exact**: ahrefs.com/api/
  - **domain**: ahrefs.com/*
  - **subdomains**: ahrefs.com/*
  - **prefix**: ahrefs.com/api/*

Example of URL directory with subdomain:

- **Example URL**: apiv2.ahrefs.com
  - **exact**: apiv2.ahrefs.com
  - **domain**: apiv2.ahrefs.com/*
  - **subdomains**: apiv2.ahrefs.com/*
  - **prefix**: apiv2.ahrefs.com/*

3. **"order_by"** parameter is a character string that forces sorting of the results. Structure:
• **Structure:** "column_name:asc|desc"

• **Single column example:** "first_seen:asc" ~ this sorts results by first_seen column in ascending order

• **Multi column example:** "last_seen:desc,first_seen:asc" ~ this sorts results by 1) last_seen column in descending order, and next by 2) first_seen column in ascending order

4. "where" & "having" are **EXPERIMENTAL** parameters of condition sets (character strings) that control filtering the results. To create arguments:

1. use `rah_condition()` function to create a single condition, for example: `cond_1 <- rah_condition(column = "links", operator = "GREATER_THAN", value = "10")`

2. use `rah_condition_set()` function to group single conditions into final condition string, for example: `fin_cond <- rah_condition_set(cond_1, cond_2)`

3. provide final condition to proper report function as a parameter, for example: `RAhrefs::rah_pages_info(target = "ahrefs.com", token = "0123456789", mode = "domain", metrics = NULL, limit = 1000, where = fin_cond, order_by = "first_seen:asc")`

**Value**

nested list - the structure can be too complicated to convert into simple data frame

**Source**

https://ahrefs.com/api/documentation

**See Also**

Other Ahrefs reports: `rah_aahrefs_rank, rah_anchors_refdomains, rah_anchors, rah_backlinks_new_lost_counters, rah_backlinks_new_lost, rah_backlinks_one_per_domain, rah_backlinks, rah_broken_backlinks, rah_broken_links, rah_domain_rating, rah_linked_anchors, rah_linked_domains_by_type, rah_linked_domains, rah_metrics_extended, rah_metrics, rah_pages_extended, rah_pages, rah_refdomains_by_type, rah_refdomains_new_lost, rah_refdomains_new_lost_counters, rah_refdomains, rah_refips, rah_subscription_info`

**Examples**

```r
# Not run:
# creating single conditions for 'where' parameter
cond_1 <- RAhrefs::rah_condition(
  column_name = "facebook_likes",
  operator = "GREATER_OR_EQUAL",
  value = "1000")

cond_2 <- RAhrefs::rah_condition(
  column_name = "facebook_shares",
  operator = "GREATER_THAN",
  value = "200")

# joining conditions into one condition set
cond_where <- RAhrefs::rah_condition_set(cond_1, cond_2)
```
# downloading
b <- RAhrefs::rah_pages_info(
  target = "ahrefs.com",
  limit = 2,
  where = cond_where,
  order_by = "ahrefs_rank:desc")

## End(Not run)

##

### rah_refdomains

Export the referring domains that contain backlinks to the target.

**Description**

Export the referring domains that contain backlinks to the target.

**Usage**

```r
rah_refdomains(target, token = Sys.getenv("AHREFS_AUTH_TOKEN"),
  mode = "domain", metrics = NULL, limit = 1000, order_by = NULL,
  where = NULL, having = NULL)
```

**Arguments**

- `target`: character string. Aim of a request: a domain, a directory or a URL.
- `token`: character string. Authentication token. Should be available through enviromental variables after authentication with function `rah_auth()`.
- `mode`: character string. Mode of operation: exact, domain, subdomains or prefix. See more in Details section.
- `metrics`: character vector of columns to select. See more in Details section.
- `limit`: integer. Number of results to return.
- `order_by`: character vector of columns to sort on. See more in Details section.
- `where`: character string - a condition created by `rah_condition_set()` function that generates proper "where" condition to satisfy. See more in Details section.
- `having`: character string - a condition created by `rah_condition_set()` function that generates proper "having" condition to satisfy. See more in Details section.

**Details**

1. **available metrics** - you can select which columns (metrics) you want to download and which one would be useful in filtering, **BUT not all of them can always be used** in "where" & "having" conditions:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Where</th>
<th>Having</th>
<th>Description</th>
</tr>
</thead>
</table>


url        string    +    -    Target of the request.
ip        string    +    -    IP address of the referring domain that links to the target.
refdomain  string    +    +    The referring domain that contains at least one link to the target.
backlinks  int    +    +    Number of backlinks found in the referring domain that link to the target.
refpages   int    +    +    Number of referring pages found in the referring domain that link to the target.
first_seen  date   +    +    Least recent date when the Ahrefs crawler was able to visit the backlinks in the referring domain.
last_visited  date   +    +    Most recent date when the Ahrefs crawler was able to visit the backlinks in the referring domain.
domain_rating  int   -    +    Domain Rating of the referring domain.

2. "mode" parameter can take 4 different values that will affect how the results will be grouped.
   Example of URL directory with folder:
   • Example URL: ahrefs.com/api/
     • exact: ahrefs.com/api/
     • domain: ahrefs.com/*
     • subdomains: *ahrefs.com/*
     • prefix: ahrefs.com/api/*

   Example of URL directory with subdomain:
   • Example URL: apiv2.ahrefs.com
     • exact: apiv2.ahrefs.com/
     • domain: apiv2.ahrefs.com/*
     • subdomains: *apiv2.ahrefs.com/*
     • prefix: apiv2.ahrefs.com/*

3. "order_by" parameter is a character string that forces sorting of the results. Structure:
   • Structure: "column_name:asc|desc"
   • Single column example: "first_seen:asc" ~ this sorts results by first_seen column in ascending order
   • Multi column example: "last_seen:desc,first_seen:asc" ~ this sorts results by 1) last_seen column in descending order, and next by 2) first_seen column in ascending order

4. "where" & "having" are EXPERIMENTAL parameters of condition sets (character strings) that control filtering the results. To create arguments:
   1. use rah_condition() function to create a single condition, for example: cond_1<-rah_condition(column = "links", operator = "GREATER_THAN", value = "10")
   2. use rah_condition_set() function to group single conditions into final condition string, for example: fin_cond <- rah_condition_set(cond_1, cond_2)
   3. provide final condition to proper report function as a parameter, for example: RAhrefs::rah_refdomains(target = "ahrefs.com", report = "anchors", token = "0123456789", mode = "domain", metrics = NULL, limit = 1000, where = fin_cond, order_by = "first_seen:asc")

Value
   data frame
### Export the referring domains that contain backlinks to the target.

**Description**

Export the referring domains that contain backlinks to the target.
**Usage**

`rah_refdomains_by_type(target, token = Sys.getenv("AHREFS_AUTH_TOKEN"),
mode = "domain", metrics = NULL, limit = 1000, order_by = NULL,
where = NULL, having = NULL)`

**Arguments**

- **target**: character string. Aim of a request: a domain, a directory or a URL.
- **token**: character string. Authentication token. Should be available through environmental variables after authentication with function `rah_auth()`.
- **mode**: character string. Mode of operation: exact, domain, subdomains or prefix. See more in Details section.
- **metrics**: character vector of columns to select. See more in Details section.
- **limit**: integer. Number of results to return.
- **order_by**: character vector of columns to sort on. See more in Details section.
- **where**: character string - a condition created by `rah_condition_set()` function that generates proper "where" condition to satisfy. See more in Details section.
- **having**: character string - a condition created by `rah_condition_set()` function that generates proper "having" condition to satisfy. See more in Details section.

**Details**

1. **available metrics** - you can select which columns (metrics) you want to download and which one would be useful in filtering, **BUT not all of them can always be used** in "where" & "having" conditions:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Where</th>
<th>Having</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>string</td>
<td>+</td>
<td>-</td>
<td>Target of the request.</td>
</tr>
<tr>
<td>ip</td>
<td>string</td>
<td>+</td>
<td>-</td>
<td>IP address of the referring domain that links to the target.</td>
</tr>
<tr>
<td>refdomain</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>The referring domain that contains at least one link to the target.</td>
</tr>
<tr>
<td>all</td>
<td>bool</td>
<td>+</td>
<td>-</td>
<td>Set to true for referring domains that contain at least one backlink to the target.</td>
</tr>
<tr>
<td>text</td>
<td>bool</td>
<td>+</td>
<td>-</td>
<td>Set to true for referring domains that contain at least one text backlink to the target.</td>
</tr>
<tr>
<td>image</td>
<td>bool</td>
<td>+</td>
<td>-</td>
<td>Set to true for referring domains that contain at least one image backlink to the target.</td>
</tr>
<tr>
<td>nofollow</td>
<td>bool</td>
<td>+</td>
<td>-</td>
<td>Set to true for referring domains that contain at least one nofollow backlink to the target.</td>
</tr>
<tr>
<td>dofollow</td>
<td>bool</td>
<td>+</td>
<td>-</td>
<td>Set to true for referring domains that contain at least one dofollow backlink to the target.</td>
</tr>
<tr>
<td>redirect</td>
<td>bool</td>
<td>+</td>
<td>-</td>
<td>Set to true for referring domains that contain at least one redirect backlink to the target.</td>
</tr>
<tr>
<td>canonical</td>
<td>bool</td>
<td>+</td>
<td>-</td>
<td>Set to true for referring domains that contain at least one canonical backlink to the target.</td>
</tr>
<tr>
<td>gov</td>
<td>bool</td>
<td>+</td>
<td>-</td>
<td>Set to true for referring domains that contain at least one backlink to the target.</td>
</tr>
<tr>
<td>edu</td>
<td>bool</td>
<td>+</td>
<td>-</td>
<td>Set to true for referring domains that contain at least one backlink to the target.</td>
</tr>
<tr>
<td>backlinks</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Number of backlinks found in the referring domain that link to the target.</td>
</tr>
<tr>
<td>backlinks_dofollow</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Number of dofollow backlinks found in the referring domain that link to the target.</td>
</tr>
<tr>
<td>refpages</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>Number of referring pages found in the referring domain that link to the target.</td>
</tr>
<tr>
<td>first_seen</td>
<td>date</td>
<td>+</td>
<td>+</td>
<td>Least recent date when the Ahrefs crawler was able to visit the backlinks in.</td>
</tr>
<tr>
<td>last_visited</td>
<td>date</td>
<td>+</td>
<td>+</td>
<td>Most recent date when the Ahrefs crawler was able to visit the backlinks in.</td>
</tr>
<tr>
<td>domain_rating</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>Domain Rating of the referring domain.</td>
</tr>
</tbody>
</table>
2. "mode" parameter can take 4 different values that will affect how the results will be grouped.

Example of URL directory with folder:

- **Example URL:** ahrefs.com/api/
- **exact:** ahrefs.com/api/
- **domain:** ahrefs.com/*
- **subdomains:** *ahrefs.com/*
- **prefix:** ahrefs.com/api/*

Example of URL directory with subdomain:

- **Example URL:** apiv2.ahrefs.com
- **exact:** apiv2.ahrefs.com/
- **domain:** apiv2.ahrefs.com/*
- **subdomains:** *apiv2.ahrefs.com/*
- **prefix:** apiv2.ahrefs.com/*

3. "order_by" parameter is a character string that forces sorting of the results. Structure:

- **Structure:** "column_name:asc|desc"
- **Single column example:** "first_seen:asc" - this sorts results by first_seen column in ascending order
- **Multi column example:** "last_seen:desc,first_seen:asc" - this sorts results by 1) last_seen column in descending order, and next by 2) first_seen column in ascending order

4. "where" & "having" are EXPERIMENTAL parameters of condition sets (character strings) that control filtering the results. To create arguments:

1. use `rah_condition()` function to create a single condition, for example: `cond_1 <- rah_condition(column = "links", operator = "GREATER_THAN", value = "10")`
2. use `rah_condition_set()` function to group single conditions into final condition string, for example: `fin_cond <- rah_condition_set(cond_1, cond_2)`
3. provide final condition to proper report function as a parameter, for example: `RAhrefs::rah_refdomains_by_type(target = "ahrefs.com", token = "0123456789", mode = "domain", metrics = NULL, limit = 1000, where = fin_cond, order_by = "first_seen:asc")`

**Value**

data frame

**Source**

https://ahrefs.com/api/documentation
See Also

Other Ahrefs reports: rah_ahrefs_rank, rah_anchors_refdomains, rah_anchors, rah_backlinks_new_lost_counters, rah_backlinks_new_lost, rah_backlinks_one_per_domain, rah_backlinks, rah_broken_backlinks, rah_broken_links, rah_domain_rating, rah_linked_anchors, rah_linked_domains_by_type, rah_linked_domains, rah_metrics_extended, rah_metrics, rah_pages_extended, rah_pages_info, rah_pages, rah_refdomains_new_lost_counters, rah_refdomains_new_lost, rah_refdomains, rah_refips, rah_subscription_info

Examples

```r
## Not run:
# creating single conditions for 'where' parameter
cond_1 <- RAhrefs::rah_condition(
    column_name = "backlinks",
    operator = "GREATER_OR_EQUAL",
    value = "20")

cond_2 <- RAhrefs::rah_condition(
    column_name = "refpages",
    operator = "GREATER_THAN",
    value = "10")

# joining conditions into one condition set
cond_where <- RAhrefs::rah_condition_set(cond_1, cond_2)

# downloading
b <- RAhrefs::rah_refdomains(
    target = "ahrefs.com",
    limit = 2,
    where = cond_where,
    order_by = "ahrefs_rank:desc")

## End(Not run)
```

---

**rah_refdomains_new_lost**

Export the new or lost referring domains and their details.

### Description

Export the new or lost referring domains and their details.

### Usage

```r
rah_refdomains_new_lost(target, token = Sys.getenv("AHREFS_AUTH_TOKEN"),
    mode = "domain", metrics = NULL, limit = 1000, order_by = NULL,
    where = NULL, having = NULL)
```
Arguments

target    character string. Aim of a request: a domain, a directory or a URL
token     character string. Authentication token. Should be available through environmental variables after authentication with function rah_auth()
mode      character string. Mode of operation: exact, domain, subdomains or prefix. See more in Details section
metrics   character vector of columns to select. See more in Details section
limit     integer. Number of results to return
order_by  character vector of columns to sort on. See more in Details section
where     character string - a condition created by rah_condition_set() function that generates proper "where" condition to satisfy. See more in Details section
having    character string - a condition created by rah_condition_set() function that generates proper "having" condition to satisfy. See more in Details section

Details

1. available metrics - you can select which columns (metrics) you want to download and which one would be useful in filtering, BUT not all of them can always be used in "where" & "having" conditions:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Where</th>
<th>Having</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>date</td>
<td>date</td>
<td>+</td>
<td>+</td>
<td>Date when the Ahrefs crawler was able to visit new or lost backlinks from the referring domain.</td>
</tr>
<tr>
<td>type</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Indicates whether the backlinks from the referring domain are new or lost.</td>
</tr>
<tr>
<td>refdomain</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>Referring domain that contains at least one link to the target.</td>
</tr>
<tr>
<td>domain_rating</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>Domain Rating of the referring domain.</td>
</tr>
</tbody>
</table>

2. "mode" parameter can take 4 different values that will affect how the results will be grouped.
Example of URL directory with folder:

- **Example URL**: ahrefs.com/api/
- **exact**: ahrefs.com/api/
- **domain**: ahrefs.com/*
- **subdomains**: *ahrefs.com/*
- **prefix**: ahrefs.com/api/*

Example of URL directory with subdomain:

- **Example URL**: apiv2.ahrefs.com
- **exact**: apiv2.ahrefs.com/
- **domain**: apiv2.ahrefs.com/*
- **subdomains**: *apiv2.ahrefs.com/*
- **prefix**: apiv2.ahrefs.com/*
3. "order_by" parameter is a character string that forces sorting of the results. Structure:

- Structure: "column_name:asc|desc"
- Single column example: "first_seen:asc" ~ this sorts results by first_seen column in ascending order
- Multi column example: "last_seen:desc,first_seen:asc" ~ this sorts results by 1) last_seen column in descending order, and next by 2) first_seen column in ascending order

4. "where" & "having" are EXPERIMENTAL parameters of condition sets (character strings) that control filtering the results. To create arguments:

1. use `rah_condition()` function to create a single condition, for example:
   ```r
   cond_1 <- rah_condition(column_name = "links", operator = "GREATER_THAN", value = "10")
   ```

2. use `rah_condition_set()` function to group single conditions into final condition string, for example:

   ```r
   fin_cond <- rah_condition_set(cond_1, cond_2)
   ```

3. provide final condition to proper report function as a parameter, for example:

   ```r
   RAhrefs::rah_refdomains_new_lost(target = "ahrefs.com", token = "0123456789", mode = "domain", metrics = NULL, limit = 1000, where = fin_cond, order_by = "first_seen:asc")
   ```

Value
data frame

Source

https://ahrefs.com/api/documentation

See Also

Other Ahrefs reports: `rah_ahrefs_rank`, `rah_anchors_refdomains`, `rah_anchors`, `rah_backlinks_new_lost_counters`, `rah_backlinks_new_lost`, `rah_backlinks_one_per_domain`, `rah_backlinks`, `rah_broken_backlinks`, `rah_domain_rating`, `rah_linked_anchors`, `rah_linked_domains_by_type`, `rah_linked_domains`, `rah_metrics_extended`, `rah_metrics`, `rah_pages_extended`, `rah_pages_info`, `rah_pages`, `rah_refdomains_by_type`, `rah_refdomains_new_lost_counters`, `rah_refdomains`, `rah_refips`, `rah_subscription_info`

Examples

```r
## Not run:
# downloading
b <- RAhrefs::rah_refdomains_new_lost(
   target = "ahrefs.com",
   limit = 2,
   order_by = "domain_rating:desc")

## End(Not run)
```
**Export new and lost domains totals.**

**Description**

Export new and lost domains totals.

**Usage**

```r
rah_refdomains_new_lost_counters(target,
    token = Sys.getenv("AHREFS_AUTH_TOKEN"), mode = "domain",
    metrics = NULL, limit = 1000, order_by = NULL, where = NULL,
    having = NULL)
```

**Arguments**

- **target** character string. Aim of a request: a domain, a directory or a URL.
- **token** character string. Authentication token. Should be available through enviromental variables after authentication with function `rah_auth()`.
- **mode** character string. Mode of operation: exact, domain, subdomains or prefix. See more in Details section.
- **metrics** character vector of columns to select. See more in Details section.
- **limit** integer. Number of results to return.
- **order_by** character vector of columns to sort on. See more in Details section.
- **where** character string - a condition created by `rah_condition_set()` function that generates proper "where" condition to satisfy. See more in Details section.
- **having** character string - a condition created by `rah_condition_set()` function that generates proper "having" condition to satisfy. See more in Details section.

**Details**

1. **available metrics** - you can select which columns (metrics) you want to download and which one would be useful in filtering, **BUT not all of them can always be used** in "where" & "having" conditions:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Where</th>
<th>Having</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>date</td>
<td>date</td>
<td>+</td>
<td>+</td>
<td>Date when the Ahrefs crawler was able to visit new or lost backlinks from the referring domain.</td>
</tr>
<tr>
<td>type</td>
<td>string</td>
<td>+</td>
<td>-</td>
<td>Indicates whether the backlinks from the referring domain are new or lost.</td>
</tr>
<tr>
<td>refdomain</td>
<td>string</td>
<td>+</td>
<td>-</td>
<td>Referring domain that contains at least one link to the target.</td>
</tr>
<tr>
<td>new</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>Total number of new referring domain.</td>
</tr>
<tr>
<td>lost</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>Total number of lost referring domain.</td>
</tr>
<tr>
<td>new_total</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>Total number of new referring domain when ignoring where filter.</td>
</tr>
<tr>
<td>lost_total</td>
<td>int</td>
<td>-</td>
<td>+</td>
<td>Total number of lost referring domain when ignoring where filter.</td>
</tr>
</tbody>
</table>
2. "mode" parameter can take 4 different values that will affect how the results will be grouped.

Example of URL directory with folder:

- **Example URL:** ahrefs.com/api/
- **exact:** ahrefs.com/api/
- **domain:** ahrefs.com/*
- **subdomains:** *ahrefs.com/*
- **prefix:** ahrefs.com/api/*

Example of URL directory with subdomain:

- **Example URL:** apiv2.ahrefs.com
- **exact:** apiv2.ahrefs.com/
- **domain:** apiv2.ahrefs.com/*
- **subdomains:** *apiv2.ahrefs.com/*
- **prefix:** apiv2.ahrefs.com/*

3. "order_by" parameter is a character string that forces sorting of the results. Structure:

- **Structure:** "column_name:asc|desc"
- **Single column example:** "first_seen:asc" - this sorts results by first_seen column in ascending order
- **Multi column example:** "last_seen:desc,first_seen:asc" - this sorts results by 1) last_seen column in descending order, and next by 2) first_seen column in ascending order

4. "where" & "having" are EXPERIMENTAL parameters of condition sets (character strings) that control filtering the results. To create arguments:

1. use `rah_condition()` function to create a single condition, for example: `cond_1 <- rah_condition(column = "links", operator = "GREATER_THAN", value = "10")`
2. use `rah_condition_set()` function to group single conditions into final condition string, for example: `fin_cond <- rah_condition_set(cond_1, cond_2)`
3. provide final condition to proper report function as a parameter, for example: `RAhrefs::rah_refdomains_new_lost_counters(target = "ahrefs.com", token = "0123456789", mode = "domain", metrics = NULL, limit = 1000, where = fin_cond, order_by = "first_seen:asc")`

**Value**

data frame

**Source**

https://ahrefs.com/api/documentation
### Examples

```r
## Not run:
# downloading
b <- RAhrefs::rah_refdomains_new_lost_counters(
  target = "ahrefs.com",
  limit = 2,
  order_by = "new_total:desc")
## End(Not run)
```

### Description

Export the referring IP addresses that have at least one link to the target.

### Usage

```r
rah_refips(target, token = Sys.getenv("AHREFS_AUTH_TOKEN"),
    mode = "domain", metrics = NULL, limit = 1000, order_by = NULL,
    where = NULL, having = NULL)
```

### Arguments

- **target**: character string. Aim of a request: a domain, a directory or a URL.
- **token**: character string. Authentication token. Should be available through environmental variables after authentication with function `rah_auth()`.
- **mode**: character string. Mode of operation: exact, domain, subdomains or prefix. See more in Details section.
- **metrics**: character vector of columns to select. See more in Details section.
- **limit**: integer. Number of results to return.
- **order_by**: character vector of columns to sort on. See more in Details section.
- **where**: character string - a condition created by `rah_condition_set()` function that generates proper "where" condition to satisfy. See more in Details section.
- **having**: character string - a condition created by `rah_condition_set()` function that generates proper "having" condition to satisfy. See more in Details section.
Details

1. **available metrics** - you can select which columns (metrics) you want to download and which one would be useful in filtering, **BUT not all of them can always be used** in "where" & "having" conditions:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Where</th>
<th>Having</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>refip</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>The referring IP address that links to the target.</td>
</tr>
<tr>
<td>refdomain</td>
<td>string</td>
<td>+</td>
<td>+</td>
<td>The domain name for the referring IP address.</td>
</tr>
<tr>
<td>backlinks</td>
<td>int</td>
<td>+</td>
<td>+</td>
<td>The number of backlinks from the referring IP address with particular domain name.</td>
</tr>
</tbody>
</table>

2. **"mode"** parameter can take 4 different values that will affect how the results will be grouped.

Example of URL directory with folder:

- Example URL: ahrefs.com/api/
  - exact: ahrefs.com/api/
  - domain: ahrefs.com/*
  - subdomains: *ahrefs.com/*
  - prefix: ahrefs.com/api/*

Example of URL directory with subdomain:

- Example URL: apiv2.ahrefs.com
  - exact: apiv2.ahrefs.com/
  - domain: apiv2.ahrefs.com/*
  - subdomains: *apiv2.ahrefs.com/*
  - prefix: apiv2.ahrefs.com/*

3. **"order_by"** parameter is a character string that forces sorting of the results. Structure:

- Structure: "column_name:asc|desc"
- Single column example: "first_seen:asc" ~ this sorts results by first_seen column in ascending order
- Multi column example: "last_seen:desc,first_seen:asc" ~ this sorts results by 1) last_seen column in descending order, and next by 2) first_seen column in ascending order

4. "where" & "having" are **EXPERIMENTAL** parameters of condition sets (character strings) that control filtering the results. To create arguments:

1. use `rah_condition()` function to create a single condition, for example: `cond_1 <- rah_condition(column = "links", operator = "GREATER_THAN", value = "10")`
2. use `rah_condition_set()` function to group single conditions into final condition string, for example: `fin_cond <- rah_condition_set(cond_1, cond_2)`
3. provide final condition to proper report function as a parameter, for example: `RAhrefs::rah_refips(target = "ahrefs.com", token = "0123456789", mode = "domain", metrics = NULL, limit = 1000, where = fin_cond, order_by = "first_seen:asc")`
rah_subscription_info

Value
data frame

Source
https://ahrefs.com/api/documentation

See Also
Other Ahrefs reports: rah_ahrefs_rank, rah_anchors_refdomains, rah_anchors, rah_backlinks_new_lost_counters, rah_backlinks_new_lost, rah_backlinks_one_per_domain, rah_backlinks, rahroken_backlinks, rah_domain_rating, rah_linked_anchors, rah_linked_domains_by_type, rah_linked_domains, rah_metrics_extended, rah_metrics, rah_pages_extended, rah_pages_info, rah_pages, rah_refdomains_by_type, rah_refdomains_new_lost_counters, rah_refdomains_new_lost, rah_refdomains, rah_subscription_info

Examples

```r
## Not run:
# downloading
b <- RAhrefs::rah_refips(
  target = "ahrefs.com",
  limit = 2,
  order_by = "backlinks:desc")

## End(Not run)
```

---

rah_subscription_info

Export user subscription information.

Description

Export user subscription information.

Usage

```r
rah_subscription_info(target, token = Sys.getenv("AHREFS_AUTH_TOKEN"),
  mode = "domain", metrics = NULL, limit = 1000, order_by = NULL,
  where = NULL, having = NULL)
```

Arguments

- `target` character string. Aim of a request: a domain, a directory or a URL
- `token` character string. Authentication token. Should be available through enviromental variables after authentication with function `rah_auth()`
mode | character string. Mode of operation: exact, domain, subdomains or prefix. See more in Details section
metrics | character vector of columns to select. See more in Details section
limit | integer. Number of results to return
order_by | character vector of columns to sort on. See more in Details section
where | character string - a condition created by `rah_condition_set()` function that generates proper "where" condition to satisfy. See more in Details section
having | character string - a condition created by `rah_condition_set()` function that generates proper "having" condition to satisfy. See more in Details section

Details

1. available metrics - you can select which columns (metrics) you want to download and which one would be useful in filtering, BUT not all of them can always be used in "where" & "having" conditions:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Where</th>
<th>Having</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rows_left</td>
<td>int</td>
<td>-</td>
<td>-</td>
<td>Number of rows available for making API requests.</td>
</tr>
<tr>
<td>rows_limit</td>
<td>int</td>
<td>-</td>
<td>-</td>
<td>Total number of rows available for the subscription plan.</td>
</tr>
<tr>
<td>subscription</td>
<td>string</td>
<td>-</td>
<td>-</td>
<td>Name of the API subscription plan.</td>
</tr>
</tbody>
</table>

2. "mode" parameter can take 4 different values that will affect how the results will be grouped.

Example of URL directory with folder:

- **Example URL**: ahrefs.com/api/
- **exact**: ahrefs.com/api/
- **domain**: ahrefs.com/*
- **subdomains**: *ahrefs.com/*
- **prefix**: ahrefs.com/api/*

Example of URL directory with subdomain:

- **Example URL**: apiv2.ahrefs.com
- **exact**: apiv2.ahrefs.com/
- **domain**: apiv2.ahrefs.com/*
- **subdomains**: *apiv2.ahrefs.com/*
- **prefix**: apiv2.ahrefs.com/*

3. "order_by" parameter is a character string that forces sorting of the results. Structure:

- **Structure**: "column_name:asc|desc"
- **Single column example**: "first_seen:asc" ~ this sorts results by first_seen column in ascending order
- **Multi column example**: "last_seen:desc,first_seen:asc" ~ this sorts results by 1) last_seen column in descending order, and next by 2) first_seen column in ascending order
4. "where" & "having" are **EXPERIMENTAL** parameters of condition sets (character strings) that control filtering the results. To create arguments:

1. use `rah_condition()` function to create a single condition, for example:
   ```r
   cond_1 <- rah_condition(column = "links", operator = "GREATER_THAN", value = "10")
   ```

2. use `rah_condition_set()` function to group single conditions into final condition string, for example:
   ```r
   fin_cond <- rah_condition_set(cond_1, cond_2)
   ```

3. provide final condition to proper report function as a parameter, for example:
   ```r
   RAhrefs::rah_subscription_info()
   ```

**Value**

data frame

**Source**

https://ahrefs.com/api/documentation

**See Also**

Other Ahrefs reports: `rah_a hrefs_rank`, `rah_anchors_refdomains`, `rah_anchors`, `rah_backlinks_new_lost_counts`, `rah_backlinks`, `rah_backlinks_one_per_domain`, `rah_backlinks`, `rah_broken_backlinks`, `rah_broken_links`, `rah_domain_rating`, `rah_linked_anchors`, `rah_linked_domains_by_type`, `rah_linked_domains`, `rah_metrics_extended`, `rah_metrics`, `rah_pages_extended`, `rah_pages_info`, `rah_pages`, `rah_refdomains_by_type`, `rah_refdomains_new_lost_counts`, `rah_refdomains_new_lost`, `rah_refdomains`, `rah_refips`

**Examples**

```r
## Not run:
# downloading
b <- RAhrefs::rah_subscription_info()
```

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