

# 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](mailto: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

rah_backlinks_new_lost . . . . .	14
rah_backlinks_new_lost_counters . . . . .	17
rah_backlinks_one_per_domain . . . . .	20
rah_broken_backlinks . . . . .	23
rah_broken_links . . . . .	26
rah_condition . . . . .	29
rah_condition_set . . . . .	31
rah_domain_rating . . . . .	32
rah_downloader . . . . .	34
rah_linked_anchors . . . . .	36
rah_linked_domains . . . . .	39
rah_linked_domains_by_type . . . . .	42
rah_metrics . . . . .	45
rah_metrics_extended . . . . .	47
rah_pages . . . . .	50
rah_pages_extended . . . . .	52
rah_pages_info . . . . .	55
rah_refdomains . . . . .	58
rah_refdomains_by_type . . . . .	61
rah_refdomains_new_lost . . . . .	64
rah_refdomains_new_lost_counters . . . . .	66
rah_refips . . . . .	68
rah_subscription_info . . . . .	71

---

ahrefs_metrics	<i>Ahrefs metrics</i>
----------------	-----------------------

---

## Description

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

## Usage

```
data(ahrefs_metrics)
```

## Format

An object of class 'data.frame'

## Source

Ahrefs API Documentation<sup>1</sup>

## Examples

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

---

<sup>1</sup><https://ahrefs.com/api/documentation>

---

ahrefs\_reports      *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<sup>2</sup>

### 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)
```

---

<sup>2</sup><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 enviromental variables after authentication with function <code>rah_auth()</code>
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 <code>rah_condition_set()</code> function that generates proper "where" condition to satisfy. See more in Details section
having	character string - a condition created by <code>rah_condition_set()</code> 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:

Column	Type	Where	Having	Description
url	string	+	+	Target of the request.
ahrefs_rank	int	-	+	URL Rating of 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/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

```
## 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	<i>Export the anchor text and the number of backlinks, referring pages and referring domains that has it.</i>
-------------	---

---

## 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 enviromental variables after authentication with function <code>rah_auth()</code>
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 <code>rah_condition_set()</code> function that generates proper "where" condition to satisfy. See more in Details section
having	character string - a condition created by <code>rah_condition_set()</code> 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:

Column	Type	Where	Having	Description
anchor	string	+	+	Anchor text used in at least one backlink from the referring domain.
backlinks	int	+	+	Number of external backlinks found that are using the anchor text.
refpages	int	+	+	Number of pages found containing backlinks that are using the anchor text.
refdomain	string	+	-	Referring domain that contains at least one backlink using the anchor text.
refdomains	int	-	+	Number of referring domains that are using the anchor text when linking to the target.
first_seen	date	+	+	Least recent date when the Ahrefs crawler was able to visit the backlink that is using the anchor text.
last_visited	date	+	+	Most recent date when the Ahrefs crawler was able to visit the backlink that is using 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: `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 <code>rah_auth()</code>
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 <code>rah_condition_set()</code> function that generates proper "where" condition to satisfy. See more in Details section
having	character string - a condition created by <code>rah_condition_set()</code> 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:

Column	Type	Where	Having	Description
anchor	string	+	-	Anchor text used in at least one backlink from the referring domain.
anchors	int	-	+	Number of different anchor texts that are used in backlinks to the target on the refer
backlinks	int	+	+	Number of external backlinks found that are using the anchor text.
refdomain	string	+	+	Referring domain that contains at least one backlink using 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:ascldesc"
- **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("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\_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
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**

```
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**

```
## Not run:
rah_auth("ABCDEFGHJKLMNOPQRST")

## End(Not run)
```

---

rah_backlinks	<i>Export the backlinks and details of the referring pages, such as anchor and page title.</i>
---------------	--

---

**Description**

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

**Usage**

```
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 <code>rah_auth()</code>
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 <code>rah_condition_set()</code> function that generates proper "where" condition to satisfy. See more in Details section
having	character string - a condition created by <code>rah_condition_set()</code> 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:

Column	Type	Where	Having	Description
url_from	string	+	+	URL of the page where the backlink is found.
url_to	string	+	+	URL of the page the backlink is pointing to.
ahrefs_rank	int	+	+	URL Rating of the referring page.
domain_rating	int	-	+	Domain Rating of the referring domain.
ahrefs_top	int	-	+	Ahrefs Rank of the target domain.
ip_from	string	+	+	IP address of the referring page.
links_internal	int	+	+	Number of internal links found in the referring page.
links_external	int	+	+	Number of external links found in the referring page.
page_size	int	+	+	Size of the referring page, in bytes.
encoding	string	+	+	Character encoding of the referring page, for example "utf8" or "iso-8859-1".
language	string	+	+	Language of the referring page (ISO 639-1).
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 crawler first visited the page.
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.
anchor	string	+	+	Anchor text of the backlink.
text_pre	string	+	+	Snippet before the anchor text.
text_post	string	+	+	Snippet after the anchor text.
http_code	int	+	+	The HTTP code for the Link URL.
url_from_first_seen	date	+	+	Least recent date when the Ahrefs crawler was able to visit the referring page.

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_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

```
## 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,
```

```

where      = cond_where,
order_by  = "ahrefs_rank:desc")

## End(Not run)

```

---

```
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

```
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 <code>rah_auth()</code>
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 <code>rah_condition_set()</code> function that generates proper "where" condition to satisfy. See more in Details section
having	character string - a condition created by <code>rah_condition_set()</code> 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:

Column	Type	Where	Having	Description
date	date	+	+	Date the backlink was tagged as New or Lost.
type	string	+	+	Indicates whether the backlink is New or Lost.
ahrefs_rank	int	+	+	URL Rating of the referring page.
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"
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	+	+	Date when the Ahrefs crawler was able to visit the backlink for the first time.
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 backlinks was present on the referring page when the Ahrefs crawler was able to visit the backlink.
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.
anchor	string	+	+	Anchor text of the backlink.
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: `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_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_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 = "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,
```

```

where      = cond_where,
order_by  = "domain_rating:desc")

## End(Not run)

```

---

```
rah_backlinks_new_lost_counters
```

*Export new and lost backlink totals.*

---

## Description

Export new and lost backlink totals.

## Usage

```
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 <code>rah_auth()</code>
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 <code>rah_condition_set()</code> function that generates proper "where" condition to satisfy. See more in Details section
having	character string - a condition created by <code>rah_condition_set()</code> 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:

Column	Type	Where	Having	Description
date	date	+	+	Date the backlink was tagged as New or Lost
type	string	+	-	Indicates whether the backlink is New or Lost.
ahrefs_rank	int	+	-	URL Rating of the referring page.

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"
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 was able to visit the backlink.
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.
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_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_backlinks_new_lost_counters(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\_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 = "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
```

```

b <- RAhrefs::rah_backlinks_new_lost_counters(
  target   = "ahrefs.com",
  limit    = 2,
  where    = cond_where,
  order_by = "links_external:desc")

## End(Not run)

```

---

```
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 <code>rah_auth()</code>
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 <code>rah_condition_set()</code> function that generates proper "where" condition to satisfy. See more in Details section
having	character string - a condition created by <code>rah_condition_set()</code> 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:

Column	Type	Where	Having	Description
url_from	string	+	+	URL of the page where the backlink is found.
url_to	string	+	+	URL of the page the backlink is pointing to.
ahrefs_rank	int	+	+	URL Rating of the referring page.
domain_rating	int	-	+	Domain Rating of the referring domain.
ahrefs_top	int	-	+	Ahrefs Rank of the target domain.
ip_from	string	+	+	IP address of the referring page.
links_internal	int	+	+	Number of internal links found in the referring page.
links_external	int	+	+	Number of external links found in the referring page.
page_size	int	+	+	Size of the referring page, in bytes.
encoding	string	+	+	Character encoding of the referring page, for example "utf8" or "iso-8859-1".
language	string	+	+	Language of the referring page (ISO 639-1).
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 crawler first visited the page.
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.
anchor	string	+	+	Anchor text of the backlink.
text_pre	string	+	+	Snippet before the anchor text.
text_post	string	+	+	Snippet after the anchor text.
http_code	int	+	+	The HTTP code for the Link URL.
url_from_first_seen	date	+	+	Least recent date when the Ahrefs crawler was able to visit the referring page.
total_backlinks	int	-	+	Total number of backlinks from this 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_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_backlinks_one_per_domain(token = "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\_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
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)
```

---

```
rah_broken_backlinks
```

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

---

## 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 enviromental variables after authentication with function <code>rah_auth()</code>
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 <code>rah_condition_set()</code> function that generates proper "where" condition to satisfy. See more in Details section
having	character string - a condition created by <code>rah_condition_set()</code> 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:



Column	Type	Where	Having	Description
url_from	string	+	+	URL of the page where the backlink is found.
url_to	string	+	+	URL of the page the backlink is pointing to.
ahrefs_rank	int	+	+	URL Rating of the referring page.
domain_rating	int	-	+	Domain Rating of the referring domain.
ip_from	string	+	+	IP address of the referring page.
links_internal	int	+	+	Number of internal links found in the referring page.
links_external	int	+	+	Number of external links found in the referring page.
page_size	int	+	+	Size of the referring page, in bytes.
language	string	+	+	Language of the referring page (ISO 639-1).
encoding	string	+	+	Character encoding of the referring page, for example "utf8" or "iso-8859-1"
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 b
original	boolean	+	+	Indicates whether the backlink was present on the referring page when the Ah
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.
anchor	string	+	+	Anchor text of the backlink.
text_pre	string	+	+	Snippet before the anchor text.
text_post	string	+	+	Snippet after the anchor text.
broken_at	date	+	+	The date when the Ahrefs crawler thinks the link became broken.
http_code	int	+	+	The HTTP code for the Link URL.
error	string	+	+	The string indicating the nature of error (currently only "dns").

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_broken_backlinks(column_name = "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\_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
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

```
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 <code>rah_auth()</code>
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 <code>rah_condition_set()</code> function that generates proper "where" condition to satisfy. See more in Details section
having	character string - a condition created by <code>rah_condition_set()</code> 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:

Column	Type	Where	Having	Description
url_from	string	+	+	URL of the page where the backlink is found.
url_to	string	+	+	URL of the page the backlink is pointing to.
ahrefs_rank	int	+	+	URL Rating of the referring page.
domain_rating	int	-	+	Domain Rating of the referring domain.
ip_from	string	+	+	IP address of the referring page.
links_internal	int	+	+	Number of internal links found in the referring page.
links_external	int	+	+	Number of external links found in the referring page.
page_size	int	+	+	Size of the referring page, in bytes.
encoding	string	+	+	Character encoding of the referring page, for example "utf8" or "iso-8
title	string	+	+	Title of the referring page.
language	string	+	+	Language of the referring page (ISO 639-1).
first_seen	date	+	+	Least recent date when the Ahrefs crawler was able to visit the backlin
last_visited	date	+	+	Most recent date when the Ahrefs crawler was able to visit the backlin
prev_visited	date	+	+	Second to the most recent date when the Ahrefs crawler was able to vi
original	boolean	+	+	Indicates whether the backlink was present on the referring page when
link_type	string	+	+	Either "href", "redirect", "frame", "form", "canonical", "rss", or "alter
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.
anchor	string	+	+	Anchor text of the backlink.
text_pre	string	+	+	Snippet before the anchor text.
text_post	string	+	+	Snippet after the anchor text.
broken_at	date	+	+	The date when the Ahrefs crawler thinks the link became broken.
http_code	int	+	+	The HTTP code for the Link URL.
error	string	+	+	The string indicating the nature of error (currently only "dns").
domain_to_ahrefs_top	int	-	+	Ahrefs rank of the external domain.
url_from_first_seen	date	+	+	Least recent date when the Ahrefs crawler was able to visit the referri

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_broken_links(token = "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\_counters, rah\_refdomains\_new\_lost, rah\_refdomains, rah\_refips, rah\_subscription\_info

## Examples

```
## 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	<i>Single condition creation for an Ahrefs API query</i>
---------------	--

---

## Description

This function create 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)`

**Value**

character string with formatted condition

**See Also**

Other Ahrefs conditions: rah\_condition\_set

**Examples**

```
## 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)
```

---

rah\_condition\_set *Grouping multiple conditions for an Ahrefs API query*

---

**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**

```
## 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**

Refer to **Principles of Domain Rating** (<https://help.ahrefs.com/ahrefs-metrics/how-is-ahrefs-domain-rating-calculated>) calculation for more information about Domain Rating.

**Usage**

```
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 enviromental variables after authentication with function <code>rah_auth()</code>
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 <code>rah_condition_set()</code> 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:

Column	Type	Where	Having	Description
domain_rating	int	-	+	Domain Rating of the target domain.
ahrefs_top	int	-	+	Ahrefs Rank of the target 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_domain_rating(token = "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**

```
## 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

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

## Arguments

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

- **prefix:** 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

```
# 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

```
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 enviromental variables after authentication with function <code>rah_auth()</code>
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 <code>rah_condition_set()</code> function that generates proper "where" condition to satisfy. See more in Details section
having	character string - a condition created by <code>rah_condition_set()</code> 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:

Column	Type	Where	Having	Description
anchor	string	+	+	Anchor text used in at least one outgoing link from the target domain.
links_internal	int	+	+	Number of internal outgoing links found that are using the anchor text.
links_external	int	+	+	Number of external outgoing links found that are using the anchor text.
url_from	string	+	-	URL of the page where the outgoing link is found.
url_to	string	+	-	URL of the page the outgoing link is pointing to.
ahrefs_rank	int	+	-	URL Rating of the referring page.
domain_rating	int	+	-	Domain Rating of the referring domain.
ip	string	+	-	IP address of the page.
page_size	int	+	-	Size of the referring page, in bytes.

encoding	string	+	-	Character encoding of the referring page, for example "utf8" or "iso-8859-1"
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 b
original	boolean	+	-	Indicates whether the backlink was present on the referring page when the AH
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: `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_anchors(token = "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\_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
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 enviromental variables after authentication with function <code>rah_auth()</code>
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 <code>rah_condition_set()</code> function that generates proper "where" condition to satisfy. See more in Details section
having	character string - a condition created by <code>rah_condition_set()</code> 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:

Column	Type	Where	Having	Description
domain_from	string	-	+	Target of the request.
domain_to	string	-	+	External domain that has links from the target.
links	int	-	+	Number of external links from the target that link to the external domain
unique_pages	int	-	+	Number of unique pages from the target that link to the external domain
domain_to_rating	int	-	+	Domain Rating of the external domain.
domain_to_ahrefs_top	int	-	+	Ahrefs rank of the external 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_linked_domains(token = "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**

```
## Not run:
# 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 enviromental variables after authentication with function <code>rah_auth()</code>
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:

Column	Type	Where	Having	Description
domain_from	string	+	+	Target of the request.
domain_to	string	+	+	Root of external domain that has links from the target.
links	int	-	+	Number of external links from the target that link to the external domain.
all	int	-	+	Number of external links from the target that link to the external domain.
text	int	-	+	Number of text external links from the target.
image	int	-	+	Number of image external links from the target.
nofollow	int	-	+	Number of NoFollow external links from the target.
dofollow	int	-	+	Number of DoFollow external links from the target.
redirect	int	-	+	Number of redirection external links from the target.
canonical	int	-	+	Number of canonical external links from the target.
gov	int	-	+	Number of external links from the target to governmental domain.
edu	int	-	+	Number of external links from the target to educational domain.
rss	int	-	+	Number of RSS external links from the target.
alternate	int	-	+	Number of alternate external links from the target.
first_seen	date	+	+	Least recent date when the Ahrefs crawler was able to see the link on the target.
last_visited	date	+	+	Most recent date when the Ahrefs crawler was able to see the link on the target.
domain_to_rating	int	-	+	Domain Rating of the external domain.
domain_to_ahrefs_top	int	-	+	Ahrefs rank of the external 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_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_linked_domains_by_type(column_name = "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, 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 '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)

```

---

rah_metrics	<i>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.</i>
-------------	---

---

## 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

```
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 enviromental variables after authentication with function <code>rah_auth()</code>
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 <code>rah_condition_set()</code> function that generates proper "where" condition to satisfy. See more in Details section
having	character string - a condition created by <code>rah_condition_set()</code> 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:

Column	Type	Where	Having	Description
backlinks	int	-	-	Number of external backlinks found on the referring pages that link to the target.
refpages	int	-	-	Number of external web pages containing at least one backlink that links to the target.
pages	int	-	-	Number of unique pages visited by the Ahrefs crawler on the target.
text	int	-	-	Number of backlinks that use anchor texts.
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.
gov	int	-	-	Number of backlinks of all types (including images and NoFollow) found on web pages with .gov domain.
edu	int	-	-	Number of backlinks of all types (including images and NoFollow) found on web pages with .edu domain.
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.

**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

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

## End(Not run)
```

---

```
rah_metrics_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

```
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 enviromental variables after authentication with function <code>rah_auth()</code>
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 <code>rah_condition_set()</code> function that generates proper "where" condition to satisfy. See more in Details section
having	character string - a condition created by <code>rah_condition_set()</code> 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:

Column	Type	Where	Having	Description
backlinks	int	-	-	Number of external backlinks found on the referring pages that link to the target
refpages	int	-	-	Number of external web pages containing at least one backlink that links to the target
pages	int	-	-	Number of unique pages visited by the Ahrefs crawler on the target.
valid_pages	int	-	-	Number of unique pages with non 5xx HTTP code, visited by the Ahrefs crawler
text	int	-	-	Number of backlinks that use anchor texts.



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
edu	int	-	-	Number of backlinks of all types (including images and NoFollow) found
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/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_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

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

## End(Not run)
```

---

rah\_pages

*Export the crawled pages.*

---

### Description

The result is similar to the crawled pages report found in the raw data export (<https://ahrefs.com/site-explorer/export/v2/csv/subdomains/live?target=ahrefs.com>) in Site Explorer.

### Usage

```
rah_pages(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 <code>rah_auth()</code>
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 <code>rah_condition_set()</code> function that generates proper "where" condition to satisfy. See more in Details section
having	character string - a condition created by <code>rah_condition_set()</code> 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:

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

**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(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\_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
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**

```
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 <code>rah_auth()</code>
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 <code>rah_condition_set()</code> function that generates proper "where" condition to satisfy. See more in Details section
having	character string - a condition created by <code>rah_condition_set()</code> 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:

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

**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_pages_extended(token = "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**

```
## 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	<i>Export additional info about the target, such as IP address, canonical URL, social meta tags and social metrics.</i>
----------------	---

---

### 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 <code>rah_auth()</code>
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 <code>rah_condition_set()</code> function that generates proper "where" condition to satisfy. See more in Details section
having	character string - a condition created by <code>rah_condition_set()</code> 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:

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

**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_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\_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, rah\_refdomains\_by\_type, rah\_refdomains\_new\_lost, rah\_refdomains\_new\_lost, rah\_refdomains, rah\_refips, rah\_subscription\_info

### Examples

```
## 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

```
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 <code>rah_auth()</code>
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 <code>rah_condition_set()</code> function that generates proper "where" condition to satisfy. See more in Details section
having	character string - a condition created by <code>rah_condition_set()</code> 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:

Column	Type	Where	Having	Description
--------	------	-------	--------	-------------

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 r
last_visited	date	+	+	Most recent date when the Ahrefs crawler was able to visit the backlinks in the r
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

**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\_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\_refips, rah\_subscription\_info

**Examples**

```
## 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\_by\_type

*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 enviromental variables after authentication with function <code>rah_auth()</code>
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 <code>rah_condition_set()</code> function that generates proper "where" condition to satisfy. See more in Details section
having	character string - a condition created by <code>rah_condition_set()</code> 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:

Column	Type	Where	Having	Description
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.
all	bool	+	-	Set to true for referring domains that contain at least one backlink to the target.
text	bool	+	-	Set to true for referring domains that contain at least one text backlink to the target.
image	bool	+	-	Set to true for referring domains that contain at least one image backlink to the target.
nofollow	bool	+	-	Set to true for referring domains that contain at least one nofollow backlink to the target.
dofollow	bool	+	-	Set to true for referring domains that contain at least one dofollow backlink to the target.
redirect	bool	+	-	Set to true for referring domains that contain at least one redirect backlink to the target.
canonical	bool	+	-	Set to true for referring domains that contain at least one canonical backlink to the target.
gov	bool	+	-	Set to true for referring domains that contain at least one backlink to the target.
edu	bool	+	-	Set to true for referring domains that contain at least one backlink to the target.
backlinks	int	+	+	Number of backlinks found in the referring domain that link to the target.
backlinks_dofollow	int	+	+	Number of dofollow 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_by_type("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**

```
## 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**

```
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 enviromental variables after authentication with function <code>rah_auth()</code>
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 <code>rah_condition_set()</code> function that generates proper "where" condition to satisfy. See more in Details section
having	character string - a condition created by <code>rah_condition_set()</code> 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:

Column	Type	Where	Having	Description
date	date	+	+	Date when the Ahrefs crawler was able to visit new or lost backlinks from the referring domain.
type	string	+	+	Indicates whether the backlinks from the referring domain are new or lost.
refdomain	string	+	+	Referring domain that contains at least one link to the target.
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_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\_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, rah\_refips, rah\_subscription\_info

### Examples

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

## End(Not run)
```

---

```
rah_refdomains_new_lost_counters
    Export new and lost domains totals.
```

---

## Description

Export new and lost domains totals.

## Usage

```
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 <code>rah_auth()</code>
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 <code>rah_condition_set()</code> function that generates proper "where" condition to satisfy. See more in Details section
having	character string - a condition created by <code>rah_condition_set()</code> 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:

Column	Type	Where	Having	Description
date	date	+	+	Date when the Ahrefs crawler was able to visit new or lost backlinks from the referring domain.
type	string	+	-	Indicates whether the backlinks from the referring domain are new or lost.
refdomain	string	+	-	Referring domain that contains at least one link to the target.
new	int	-	+	Total number of new referring domain.
lost	int	-	+	Total number of lost referring domain.
new_total	int	-	+	Total number of new referring domain when ignoring where filter.
lost_total	int	-	+	Total number of lost referring domain 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_refdomains_new_ = "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\_by\_type, rah\_refdomains\_new\_lost, rah\_refdomains, rah\_refips, rah\_subscription\_info

**Examples**

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

## End(Not run)
```

---

rah_refips	<i>Export the referring IP addresses that have at least one link to the target.</i>
------------	---

---

**Description**

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

**Usage**

```
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 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:

Column	Type	Where	Having	Description
refip	string	+	+	The referring IP address that links to the target.
refdomain	string	+	+	The domain name for the referring IP address.
backlinks	int	+	+	The number of backlinks from the referring IP address with particular domain name.

**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")`

**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\_by\_type, rah\_refdomains\_new\_lost\_counters, rah\_refdomains\_new\_lost, rah\_refdomains, rah\_subscription\_info

**Examples**

```
## 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**

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

Column	Type	Where	Having	Description
rows_left	int	-	-	Number of rows available for making API requests.
rows_limit	int	-	-	Total number of rows available for the subscription plan.
subscription	string	-	-	Name of the API subscription plan.

**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_subscription_in`

### 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_by_type`, `rah_refdomains_new_lost_counters`, `rah_refdomains_new_lost`, `rah_refdomains`, `rah_refips`

### Examples

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