Package 'hubspot'

August 6, 2021

Title Work with HubSpot data

Version 0.3.2

Date 2021-08-06

Description Utilising the 'Hubspot CRM' API (see
<https://developers.hubspot.com/docs> for more information), retrieve
customer relationship data for reporting and analysis.

License MIT+ file LICENSE

Depends R (>= 4.1.0)

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

Imports httr, dplyr, purrr, magrittr, tidyr, rlang, keyring, methods,
tibble, anytime, memoise, ratelimitr, glue

Suggests rmarkdown, knitr, testthat (>= 2.1.0), covr, pkgload

Author Joost Kuckartz [aut, cph, cre],
Steph Locke [aut, cph],
Jessie Geahlen [ctb],
Locke Data [fnd] (https://itsalocke.com),
Maëlle Salmon [aut] (<https://orcid.org/0000-0002-2815-0399>),
Michael W. Kearney [ctb] (<https://orcid.org/0000-0002-0730-4694>)

Maintainer Joost Kuckartz <jkuckartz1984@hotmail.com>

R topics documented:

---

hubspot-package   *hubspot*

Description

Utilizing the 'Hubspot CRM' API to retrieve customer relationship data for reporting and analysis.

Author(s)

Maintainer: Joost Kuckartz <jkuckartz1984@hotmail.com> [copyright holder]

Authors:

- Steph Locke <steph@itsalocke.com> [copyright holder]
- Maëlle Salmon <maelle.salmon@yahoo.se> (ORCID)

Other contributors:

- Jessie Geahlen <jgeahlen@gmail.com> [contributor]
- Locke Data (https://itsalocke.com) [funder]
- Michael W. Kearney <kearneymw@missouri.edu> (ORCID) [contributor]

---

hs_companies_raw   *Companies endpoint (raw and tidy)*

Description

Get raw and tidy results from the companies endpoint.

Usage

```
hs_companies_raw(
  token_path = hubspot_token_get(),
  apikey = hubspot_key_get(),
  properties = hs_company_properties_tidy(hs_company_properties_raw(token_path,
    apikey)),
  property_history = "true",
  max_iter = 10,
  max_properties = 100,
  offsetvalue = 0
)

hs_companies_tidy(companies = hs_companies_raw(), view = c("properties"))
```

Arguments

`token_path`	Path to OAuth 2.0 token. See [hubspot_token_create()]().
`apikey`	API key to work with Hubspot (only used if `token_path` is NULL). See [hubspot_key_set()]().
`properties`	Character vector of properties to request
`property_history`	
	Whether version history for properties should be returned
`max_iter`	The API is limited to a certain number of responses by page, use `max_iter` to limit how many pages of data will get returned
`max_properties`	Avoid URLs that are too long, limit the number of properties returned, if required.
`offsetvalue`	The offset value to use if continuing a previous call. Defaults to 0.
`companies`	A companies return from `hs_companies_raw()`
`view`	A view (see Details)

Details

Required scope(s) of the OAuth token: contacts.

Different `view` values and associated output.

- "properties": A tibble containing all properties of companies

Value

List with company data (`hs_companies_raw()`)

A tibble with associated entities (`hs_companies_tidy()`)

Examples

```
companies <- hs_companies_raw(
  property_history = "false", max_iter = 1,
  max_properties = 10
)
companies_properties <- hs_companies_tidy(
  companies,
  view = "properties"
)
```

---

hs_company_properties_raw

Company properties endpoint (raw and tidy)

Description

Retrieve raw and tidy results from the company properties endpoint.

Usage

```
hs_company_properties_raw(
  token_path = hubspot_token_get(),
  apikey = hubspot_key_get()
)

hs_company_properties_tidy(
  company_properties = hs_company_properties_raw(),
  view = "names"
)
```

Arguments

token_path	Path to OAuth 2.0 token. See hubspot_token_create().
apikey	API key to work with Hubspot (only used if token_path is NULL). See hubspot_key_set().
company_properties	
	Company properties as returned by hs_company_properties_raw()
view	A view (see Details)

Details

Required scope(s) of the OAuth token: contacts.

Different view values and associated output.

- "names": A character vector of the names of available company properties.

Value

List of company properties (hs_company_properties_raw())

Something tidy (hs_company_properties_tidy())

Examples

```
properties <- hs_company_properties_raw()
property_names <- hs_company_properties_tidy(properties, view = "names")
```

---

hs_contacts_raw   *Contacts endpoint (raw and tidy)*

Description

Get raw and tidy results from the contacts endpoint.

Usage

```
hs_contacts_raw(
  token_path = hubspot_token_get(),
  apikey = hubspot_key_get(),
  properties = hs_contact_properties_tidy(hs_contact_properties_raw(token_path =
    token_path, apikey = apikey)),
  property_history = "true",
  form_submission_mode = "newest",
  list_memberships = "false",
  max_iter = 10,
  max_properties = 100,
  offsetvalue = 0
)

hs_contacts_tidy(contacts = hs_contacts_raw(), view = c("properties"))
```

Arguments

token_path	Path to OAuth 2.0 token. See [hubspot_token_create()]().
apikey	API key to work with Hubspot (only used if `token_path` is NULL). See [hubspot_key_set()]().
properties	Character vector of properties to request
property_history	
	Whether version history for properties should be returned
form_submission_mode	
	Which form submissions should be fetched
list_memberships	
	Indicate whether current list memberships should be fetched
max_iter	The API is limited to a certain number of responses by page, use `max_iter` to limit how many pages of data will get returned
max_properties	Avoid URLs that are too long, limit the number of properties returned, if required.
offsetvalue	The offset value to use if continuing a previous call. Defaults to 0.
contacts	A contacts return from `hs_contacts_raw()`
view	A view (see Details)

Details

Required scope(s) of the OAuth token: contacts.

Different `view` values and associated output.

- "properties": A tibble containing all properties of contacts

Value

List with contact data (`hs_contacts_raw()`)

A tibble with associated entities (`hs_contacts_tidy()`)

Examples

```
contacts <- hs_contacts_raw(
  property_history = "false", max_iter = 1,
  max_properties = 10
)
contacts_properties <- hs_contacts_tidy(
  contacts,
  view = "properties"
)
```

---

hs_contact_lists_raw   *Contact lists endpoint (raw and tidy)*

Description

Retrieve raw and tidy results from the contact lists endpoint.

Usage

```
hs_contact_lists_raw(
  token_path = hubspot_token_get(),
  apikey = hubspot_key_get(),
  max_iter = 10,
  offsetvalue = 0
)

hs_contact_lists_tidy(
  contact_lists = hs_contact_lists_raw(),
  view = c("names")
)
```

Arguments

token_path	Path to OAuth 2.0 token. See hubspot_token_create().
apikey	API key to work with Hubspot (only used if token_path is NULL). See hubspot_key_set().
max_iter	The API is limited to a certain number of responses by page, use max_iter to limit how many pages of data will get returned
offsetvalue	The offset value to use if continuing a previous call. Defaults to 0.
contact_lists	A contact lists return from hs_contact_lists_raw()
view	A view (see Details)

Details

Required scope(s) of the OAuth token: contacts.

Different view values and associated output.

- "names": A tibble containing all IDs and names of contact lists

Value

List of contact lists (`hs_contact_lists_raw()`)

A tibble with associated entities (`hs_contact_lists_tidy()`)

Examples

```
contact_lists <- hs_contact_lists_raw()
hs_contact_lists_tidy(contact_lists)
```

---

hs_contact_properties_raw

Contact properties endpoint (raw and tidy)

Description

Retrieve raw and tidy results from the contact properties endpoint.

Usage

```
hs_contact_properties_raw(
  token_path = hubspot_token_get(),
  apikey = hubspot_key_get()
)

hs_contact_properties_tidy(
  contact_properties = hs_contact_properties_raw(),
  view = "names"
)
```

Arguments

token_path	Path to OAuth 2.0 token. See hubspot_token_create().
apikey	API key to work with Hubspot (only used if token_path is NULL). See hubspot_key_set().
contact_properties	
	Company properties as returned by hs_contact_properties_raw()
view	A view (see Details)

Details

Required scope(s) of the OAuth token: contacts.

Different `view` values and associated output.

- "names": A character vector of the names of available contact properties.

Value

List of contact properties (`hs_contact_properties_raw()`)

Something tidy (`hs_contact_properties_tidy()`)

Examples

```
properties <- hs_contact_properties_raw()
property_names <- hs_contact_properties_tidy(properties, view = "names")
```

---

hs_deals_raw   *Deals endpoint (raw and tidy)*

Description

Get raw and tidy results from the deals endpoint.

Usage

```
hs_deals_raw(
  token_path = hubspot_token_get(),
  apikey = hubspot_key_get(),
 properties = hs_deal_properties_tidy(hs_deal_properties_raw(token_path, apikey)),
  property_history = "true",
  associations = "true",
  max_iter = 10,
  max_properties = 100,
  offsetvalue = 0
)

hs_deals_tidy(
  deals = hs_deals_raw(),
  view = c("associations", "properties history", "properties", "stages history")
)
```

Arguments

token_path	Path to OAuth 2.0 token. See hubspot_token_create().
apikey	API key to work with Hubspot (only used if token_path is NULL). See hubspot_key_set().
properties	Character vector of properties to request
property_history	
	Whether version history for properties should be returned
associations	Retrieve links to other entities
max_iter	The API is limited to a certain number of responses by page, use max_iter to limit how many pages of data will get returned
max_properties	Avoid URLs that are too long, limit the number of properties returned, if required.
offsetvalue	The offset value to use if continuing a previous call. Defaults to 0.
deals	A deals list object, typically produced by running hs_deals_raw
view	A view (see Details)

Details

Required scope(s) of the OAuth token: contacts.

Different view values and associated output.

- "associations": A tibble with associated entities
- "properties history": A tibble of all field changes over time
- "properties": A tibble containing all properties of deals
- "stages history": A tibble containing all stages a deal has gone through

Value

List with deals data (hs_deals_raw())

A tibble (hs_deals_tidy())

Examples

```
deals <- hs_deals_raw(
  property_history = "false",
  max_iter = 1,
  max_properties = 10
)

deals_properties <- hs_deals_tidy(
  deals,
  view = "properties"
  )
```

---

hs_deal_pipelines_raw   *Deal pipelines endpoint (raw and tidy)*

Description

Get raw and tidy results from the deal pipelines endpoint.

Usage

```
hs_deal_pipelines_raw(
  token_path = hubspot_token_get(),
  apikey = hubspot_key_get(),
  include_inactive = "EXCLUDE_DELETED"
)

hs_deal_pipelines_tidy(
  deal_pipelines = hs_deal_pipelines_raw(),
  view = c("properties", "stages")
)
```

Arguments

token_path	Path to OAuth 2.0 token. See hubspot_token_create().
apikey	API key to work with Hubspot (only used if token_path is NULL). See hubspot_key_set().
include_inactive	
	Should inactive entities be returned
deal_pipelines	A list of pipelines from hs_deal_pipelines_raw
view	A view (see Details)

Details

Inactive pipelines are defined as deleted pipelines.

Different view values and associated output.

- "properties": tibble of pipeline IDs and property values
- "stages": tibble of pipeline IDs and stages

Value

List with deal pipeline data (hs_deal_pipelines_raw())

A tibble with associated entities (hs_deal_pipelines_tidy())

Examples

```
deal_pipelines <- hs_deal_pipelines_raw()
properties <- hs_deal_pipelines_tidy(
  deal_pipelines,
  view = "properties"
)
stages <- hs_deal_pipelines_tidy(
  deal_pipelines,
  view = "stages"
)
```

---

hs_deal_properties_raw

Deal properties endpoint (raw and tidy)

Description

Retrieve raw and tidy results from the deal properties endpoint.

Usage

```
hs_deal_properties_raw(
  token_path = hubspot_token_get(),
  apikey = hubspot_key_get()
)

hs_deal_properties_tidy(
  deal_properties = hs_deal_properties_raw(),
  view = "names"
)
```

Arguments

token_path	Path to OAuth 2.0 token. See [hubspot_token_create()](#).
apikey	API key to work with Hubspot (only used if token_path is NULL). See [hubspot_key_set()](#).
deal_properties	
	Company properties as returned by hs_deal_properties_raw()
view	A view (see Details)

Details

Required scope(s) of the OAuth token: contacts.

Different view values and associated output.

- "names": A character vector of the names of available deal properties.

Value

List of deal properties (hs_deal_properties_raw())

Something tidy (hs_deal_properties_tidy())

Examples

```
properties <- hs_deal_properties_raw()
property_names <- hs_deal_properties_tidy(properties, view = "names")
```

---

hs_engagements_raw   *Engagements endpoint (raw and tidy)*

Description

Get all of the engagements in a portal. Returns a paginated set of engagements. from the [Get all engagements endpoint](#).

Usage

```
hs_engagements_raw(
  token_path = hubspot_token_get(),
  apikey = hubspot_key_get(),
  max_iter = 10,
  offsetvalue = 0
)

hs_engagements_tidy(engagements = hs_engagements_raw(), view = "id")
```

Arguments

token_path	Path to OAuth 2.0 token. See [hubspot_token_create()](#).
apikey	API key to work with Hubspot (only used if token_path is NULL). See [hubspot_key_set()](#).
max_iter	The API is limited to a certain number of responses by page, use max_iter to limit how many pages of data will get returned
offsetvalue	The offset value to use if continuing a previous call. Defaults to 0.
engagements	An engagements return from hs_engagements_raw()
view	A view (see Details)

Details

Required scope(s) for the OAuth token: contacts.

view = "id" returns a list of all engagements, type, created and modified dates listed by ID.

view = "contacts" also returns associated contacts, one contact ID per row

view = "companies" also returns associated companies, one company ID per row

view = "deals" also returns associated deals, one deal ID per row

Value

A list (`hs_engagements_raw()`)

A tibble with associated entities (`hs_engagements_tidy()`)

Examples

```
engagements <- hs_engagements_raw()
hs_engagements_tidy(engagements)
```

---

hs_forms_from_a_portal_raw
                            WIP - Get all forms from a portal (raw and tidy)

Description

For a given portal, return all forms that have been created in the portal. from the Get all forms from a portal endpoint.

Usage

```
hs_forms_from_a_portal_raw(
  token_path = hubspot_token_get(),
  apikey = hubspot_key_get(),
  all_form_types = FALSE
)

hs_forms_from_a_portal_tidy(
  forms_from_a_portal = hs_forms_from_a_portal_raw(),
  view = "forms"
)
```

Arguments

token_path	Path to OAuth 2.0 token. See `hubspot_token_create()`.
apikey	API key to work with Hubspot (only used if `token_path` is NULL). See `hubspot_key_set()`.
all_form_types	By default non-marketing forms are filtered out of this endpoint. To request all forms, use this paramter with the value of TRUE.
forms_from_a_portal	
	A forms_from_a_portal object, typically produced by running `hs_forms_from_a_portal_raw()`
view	A view (see Details)

Details

Required scope(s) for the OAuth token: forms.

Different `view` values and associated output.

- "forms": tibble of form IDs, names, createdAt dates, upatedAt date, type, published (T/F) and number of fields
- "fields": tibble of all forms as above with form fields (one field per row)
- "options": tibble of all forms and fields as above, with options for multiple option fields (one option per row)

Value

A list (`hs_forms_from_a_portal_raw()`)

A tibble with associated entities (`hs_forms_from_a_portal_tidy()`)

Examples

```
hs_forms_from_a_portal_raw()
forms <- hs_forms_from_a_portal_tidy()
```

---

hs_lineitems_raw   *Line items endpoint (raw and tidy)*

Description

Get raw and tidy results from the line items endpoint.

Usage

```
hs_lineitems_raw(
  token_path = hubspot_token_get(),
  apikey = hubspot_key_get(),
  properties = hs_object_properties_tidy(hs_object_properties_raw("line_items",
    token_path = token_path, apikey = apikey)),
  property_history = "true",
  max_iter = 10,
  max_properties = 100,
  offsetvalue = 0
)

hs_lineitems_tidy(lineitems = hs_lineitems_raw(), view = c("properties"))
```

Arguments

token_path	Path to OAuth 2.0 token. See `hubspot_token_create()`.
apikey	API key to work with Hubspot (only used if `token_path` is NULL). See `hubspot_key_set()`.
properties	Character vector of properties to request

property_history

                Whether version history for properties should be returned

max_iter   The API is limited to a certain number of responses by page, use `max_iter` to limit how many pages of data will get returned

max_properties Avoid URLs that are too long, limit the number of properties returned, if required.

offsetvalue   The offset value to use if continuing a previous call. Defaults to 0.

lineitems   A line items return from `hs_lineitems_raw()`

view   A view (see Details)

Details

Required scope(s) of the OAuth token: e-commerce

Different `view` values and associated output.

- "properties": A tibble containing all properties of line items

Value

List with line items data (`hs_lineitems_raw()`)

A tibble with associated entities (`hs_lineitems_tidy()`)

Examples

```
lineitems <- hs_lineitems_raw(
  property_history = "false", max_iter = 1,
  max_properties = 10
)
lineitems_properties <- hs_lineitems_tidy(
  lineitems,
  view = "properties"
)
```

---

hs_object_properties_raw

                               Object properties endpoint (raw and tidy)

Description

Retrieve raw and tidy results from the object properties endpoint.

Usage

```
hs_object_properties_raw(
  objectname,
  token_path = hubspot_token_get(),
  apikey = hubspot_key_get()
)

hs_object_properties_tidy(objectname, object_properties = NULL, view = "names")
```

Arguments

`objectname`	String name of the object. Can be one of `"tickets"`, `"products"` or `"line-items"`.
`token_path`	Path to OAuth 2.0 token. See `hubspot_token_create()`.
`apikey`	API key to work with Hubspot (only used if `token_path` is `NULL`). See `hubspot_key_set()`.
`object_properties`	
	Object properties as returned by hs_object_properties_raw()
`view`	A view (see Details)

Details

Required scope(s) of the OAuth token: contacts.

Different `view` values and associated output.

- "names": A character vector of the names of available object properties.

Value

List of object properties (`hs_object_properties_raw()`)

Something tidy (`hs_object_properties_tidy()`)

Examples

```
properties <- hs_object_properties_raw("tickets")
property_names <- hs_object_properties_tidy("tickets", properties, view = "names")
```

`hs_owners_raw`	*Owners endpoint (raw and tidy)*

Description

Retrieve raw and tidy results from the owners endpoint.

Usage

```
hs_owners_raw(
  token_path = hubspot_token_get(),
  apikey = hubspot_key_get(),
  include_inactive = TRUE,
  email = NULL
)

hs_owners_tidy(owners = hs_owners_raw(), view = c("owners"))
```

Arguments

`token_path`	Path to OAuth 2.0 token. See `hubspot_token_create()`.
`apikey`	API key to work with Hubspot (only used if `token_path` is `NULL`). See `hubspot_key_set()`.
`include_inactive`	
	Should inactive entities be returned
`email`	Search for owners matching the specified email address (exact match).
`owners`	Raw data from `hs_owners_raw()`
`view`	A view (see Details)

Details

Inactive owners are defined as owners without any active remotes.

Required scope(s) of the OAuth token: contacts.

Different `view` values and associated output.

- "owners": A tibble with owners and related information.

Value

List of owners (`hs_owners_raw()`)

A tibble of owners (`hs_owners_tidy()`)

Examples

```
## Not run:
owners <- hs_owners_raw()
active_owners <- hs_owners_raw(include_inactive = FALSE)
hs_owners_raw(email = "demo@hubspot.com")

## End(Not run)
```

---

hs_products_raw   *Products endpoint (raw and tidy)*

Description

Get raw and tidy results from the products endpoint.

Usage

```
hs_products_raw(
  token_path = hubspot_token_get(),
  apikey = hubspot_key_get(),
  properties = hs_object_properties_tidy(hs_object_properties_raw("products",
    token_path = token_path, apikey = apikey)),
  property_history = "true",
  max_iter = 10,
  max_properties = 100,
  offsetvalue = 0
)

hs_products_tidy(products = hs_products_raw(), view = c("properties"))
```

Arguments

token_path	Path to OAuth 2.0 token. See hubspot_token_create().
apikey	API key to work with Hubspot (only used if `token_path` is NULL). See hubspot_key_set().
properties	Character vector of properties to request
property_history	
	Whether version history for properties should be returned

max_iter	The API is limited to a certain number of responses by page, use max_iter to limit how many pages of data will get returned
max_properties	Avoid URLs that are too long, limit the number of properties returned, if required.
offsetvalue	The offset value to use if continuing a previous call. Defaults to 0.
products	A products return from hs_products_raw()
view	A view (see Details)

Details

Required scope(s) of the OAuth token: e-commerce

Different view values and associated output.

- "properties": A tibble containing all properties of products

Value

List with products data (hs_products_raw())

A tibble with associated entities (hs_products_tidy())

Examples

```
products <- hs_products_raw(
  property_history = "false", max_iter = 1,
  max_properties = 10
)
products_properties <- hs_products_tidy(
  products,
  view = "properties"
)
```

---

hs_tickets_raw	*Tickets endpoint (raw and tidy)*

Description

Get raw and tidy results from the tickets endpoint.

Usage

```
hs_tickets_raw(
  token_path = hubspot_token_get(),
  apikey = hubspot_key_get(),
 properties = hs_object_properties_tidy(hs_object_properties_raw("tickets", token_path
    = token_path, apikey = apikey)),
  property_history = "true",
  max_iter = 10,
  max_properties = 100,
  offsetvalue = 0
)

hs_tickets_tidy(tickets = hs_tickets_raw(), view = c("properties"))
```

Arguments

`token_path`	Path to OAuth 2.0 token. See hubspot_token_create().
`apikey`	API key to work with Hubspot (only used if `token_path` is NULL). See hubspot_key_set().
`properties`	Character vector of properties to request
`property_history`	
	Whether version history for properties should be returned
`max_iter`	The API is limited to a certain number of responses by page, use `max_iter` to limit how many pages of data will get returned
`max_properties`	Avoid URLs that are too long, limit the number of properties returned, if required.
`offsetvalue`	The offset value to use if continuing a previous call. Defaults to 0.
`tickets`	A tickets return from hs_tickets_raw()
`view`	A view (see Details)

Details

Required scope(s) of the OAuth token: tickets.

Different `view` values and associated output.

- "properties": A tibble containing all properties of tickets

Value

List with tickets data (hs_tickets_raw())

A tibble with associated entities (hs_tickets_tidy())

Examples

```
tickets <- hs_tickets_raw(
  property_history = "false", max_iter = 1,
  max_properties = 10
)
tickets_properties <- hs_tickets_tidy(
  tickets,
  view = "properties"
)
```

---

hubspot_api_usage   *Check daily API usage*

Description

Check daily API usage via the dedicated API endpoint.

Usage

```
hubspot_api_usage(apikey = hubspot_key_get())
```

Arguments

apikey   API key to work with Hubspot (only used if token_path is NULL). See hubspot_key_set().

Value

A data.frame with API usage info

Examples

```
## Not run:
hubspot_api_usage()

## End(Not run)
```

---

hubspot_key_get   *Set/retrieve the Hubspot secret API key*

Description

Get/set the Hubspot secret API key using the keyring package. For a more secure authorization method, without a daily limit on calls, see hubspot_token_create()

Usage

```
hubspot_key_get()

hubspot_key_set(key = NULL)
```

Arguments

key   API key. If not provided via this argument, the key will have to be provided
                interactively.

Details

To know how to create your Hubspot API key, refer to the corresponding section of Hubspot API docs. Hubspot recommends you to re-generate your API key every six months. Only accounts with Super Admin Access can create an API key (at the moment of writing).

Save the API key on your machine using the hubspot_key_set() function, that relies on the keyring package that stores the key in the system default credential store (see keyring docs). If you don't do that, you'll have to provide the API key as argument to hubspot functions calling Hubspot APIs.

```
# Either input the key as parameter
hubspot_key_set("yourapikey")

# Or just call the function,
# the key will be asked interactively
hubspot_key_set()
```

If no API key is set nor provided as an argument, the "demo" key is used and a message is printed to the screen. As noted earlier, this won't give you access to your own Hubspot data.

You can check the daily usage related to the API key via hubspot_api_usage(). Calls in the package are rate limited (100 requests every 10 seconds) but there's also a daily limit on calls.

```
hubspot_api_usage()
```

For a more secure authorization method, without a daily limit on calls, see hubspot_token_create().

Authorization for Hubspot APIs:

The Hubspot API accepts authorization via

- an API key,
- OAuth 2.0.

OAuth 2.0 is the recommended method. However, this package supports both.

Note that if you do nothing the package will use the "demo" API token but this won't give you access to your own Hubspot data. So you'll need to spend a little time on setup:

- For rapid prototyping key, use a Hubspot API key, see hubspot_key_set().
- For more secure use, without a daily limit on API calls, see hubspot_token_create() to create a Hubspot authorization token (OAuth 2.0).

If you have both saved an API key via hubspot_key_set() and a token via hubspot_token_create(), priority will be given to using the OAuth 2.0 token. If you don't want that, explicitly pass NULL as value for the token_path argument of all functions.

Value

The key (a string)

See Also

Other auth: default_ld_hubspot_app()

Examples

```
hubspot_key_get()
## Not run:
hubspot_key_set("yourapikey")

## End(Not run)
```

Index