Package: apifetch (via r-universe)

July 2, 2026

Type Package

Title Token-Authenticated REST API Retrieval Toolkit

Version 0.1.0

Date 2026-06-26

Description A small, dependency-light toolkit for talking to token-authenticated REST APIs. It manages authentication tokens in process environment variables (never written to disk), builds requests with configurable authentication and pagination strategies, and retrieves paginated data either one page at a time or in chunks combined into a single tibble. The design is API-agnostic: a single 'apifetch_api' profile describes an endpoint together with how it authenticates and paginates, so the same verbs work across different services.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1.0)

Imports cli, dplyr, httr2, stats, tibble, utils

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

URL https://github.com/StrategicProjects/apifetch, https://strategicprojects.github.io/apifetch/

BugReports https://github.com/StrategicProjects/apifetch/issues

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author André Leite [aut, cre], Hugo Vasconcelos [aut], Diogo Bezerra [aut], Marcos Wasilew [aut], Carlos Amorin [aut]

Maintainer André Leite <leite@castlab.org>

Repository https://cran.r-universe.dev

Date/Publication 2026-07-02 18:40:19 UTC

RemoteUrl https://github.com/cran/apifetch

RemoteRef HEAD

RemoteSha 3a74d006c3b0242bc1c9acdf33af27191e4af08f

Contents

`af_api` *Describe an API endpoint*

Description

Bundles an endpoint URL with its authentication and pagination strategies and a namespace `service` (used to look up tokens). The resulting object is passed to `af_fetch()` and `af_fetch_all()`.

Usage

```
af_api(
  endpoint,
  service = "apifetch",
  auth = af_auth_bearer(),
  pagination = af_paginate_offset(),
  drop_cols = character(0),
  connect_hint = NULL
)
```

Arguments

`endpoint`	The base API URL.
`service`	Namespace used to look up the token (see `af_get_token()`).
`auth`	An `apifetch_auth` strategy (see af_auth). Default `af_auth_bearer()`.
`pagination`	An `apifetch_pagination` strategy (see af_paginate). Default `af_paginate_offset()`.

drop_cols	Character vector of response columns to drop after parsing (e.g. a status column). Default none.
connect_hint	Optional extra line shown when a connection error occurs (e.g. a VPN requirement).

Value

An apifetch_api object.

Examples

```
af_api(
  endpoint = "https://www.bigdata.pe.gov.br/api/buscar",
  service = "BigDataPE",
  auth = af_auth_raw(),
  pagination = af_paginate_offset("header"),
  drop_cols = "Mensagem",
  connect_hint = "Ensure you are on the PE Conectado network or VPN."
)
```

---

af_auth *Authentication strategies*

Description

Constructors describing how a token is attached to a request. Pass the result to the auth argument of af_api().

Usage

```
af_auth_raw(header = "Authorization")

af_auth_bearer(header = "Authorization", prefix = "Bearer ")

af_auth_header(header = "X-API-Key")

af_auth_query(param = "api_key")
```

Arguments

header	Header name to use.
prefix	String prepended to the token (bearer scheme).
param	Query-parameter name (query scheme).

Details

- `af_auth_raw()`: send the token verbatim in a header (default `Authorization`). This is what the Big Data PE API expects.
- `af_auth_bearer()`: send `"Bearer <token>"` in the `Authorization` header.
- `af_auth_header()`: send the token in an arbitrary header (e.g. `X-API-Key`).
- `af_auth_query()`: send the token as a URL query parameter.

Value

An `apifetch_auth` object.

Examples

```
af_auth_raw()
af_auth_bearer()
af_auth_header("X-API-Key")
af_auth_query("api_key")
```

---

af_fetch *Fetch a single page from an API*

Description

Performs one authenticated request against an `af_api()` profile, applying its pagination strategy, and returns the parsed body as a tibble. HTTP errors and connection failures are translated into friendly `cli` messages.

Usage

```
af_fetch(api, name, limit = Inf, offset = 0L, query = list(), verbosity = 0L)
```

Arguments

`api`	An `apifetch_api` object (see `af_api()`).
`name`	The token name to authenticate with (looked up via the API's `service`).
`limit`	Maximum number of records to request. Default `Inf` (no limit). Non-positive or infinite values omit the parameter.
`offset`	Starting record. Default `0` (omitted).
`query`	A named list of additional query-string filters. Default empty.
`verbosity`	`0` (silent, default), `1` (progress messages), or `2` (progress plus full HTTP request/response details).

Value

A tibble with the parsed response.

Examples

```
## Not run:
api <- af_api("https://www.bigdata.pe.gov.br/api/buscar",
              service = "BigDataPE", auth = af_auth_raw(),
              pagination = af_paginate_offset("header"))
af_store_token("dengue", "token", service = "BigDataPE")
af_fetch(api, "dengue", limit = 50)

## End(Not run)
```

`af_fetch_all` *Fetch all data from an API in chunks*

Description

Iteratively calls `af_fetch()` with an advancing `offset`, stopping when a chunk comes back empty or `total_limit` is reached, then row-binds the chunks into one tibble. Columns listed in the API profile's `drop_cols` are removed.

Usage

```
af_fetch_all(
  api,
  name,
  total_limit = Inf,
  chunk_size = 50000L,
  query = list(),
  verbosity = 0L
)
```

Arguments

`api`	An `apifetch_api` object (see `af_api()`).
`name`	The token name to authenticate with (looked up via the API's `service`).
`total_limit`	Maximum number of records to retrieve in total. Default `Inf` (all available).
`chunk_size`	Records to request per chunk. Default `50000`.
`query`	A named list of additional query-string filters. Default empty.
`verbosity`	`0` (silent, default), `1` (progress messages), or `2` (progress plus full HTTP request/response details).

Value

A tibble with all retrieved records.

Examples

```
## Not run:
api <- af_api("https://www.bigdata.pe.gov.br/api/buscar",
              service = "BigDataPE", auth = af_auth_raw(),
              pagination = af_paginate_offset("header"),
              drop_cols = "Mensagem")
af_fetch_all(api, "dengue", total_limit = 500, chunk_size = 100)

## End(Not run)
```

---

af_get_token *Retrieve a stored API token*

Description

Retrieves the token stored for `name` under `service`. Returns `NULL` (with a warning) rather than erroring when no token is found.

Usage

```
af_get_token(name, service = "apifetch")
```

Arguments

`name`	The identifier for this token (e.g. a dataset or resource name).
`service`	A namespace prefix grouping tokens for one API. Default `"apifetch"`.

Value

The token string, or `NULL` if not found.

Examples

```
token <- af_get_token("dengue", service = "BigDataPE")
```

---

af_list_tokens *List stored API tokens*

Description

Returns the names (without the `service` prefix) of all tokens stored for a given `service` in environment variables.

Usage

```
af_list_tokens(service = "apifetch")
```

Arguments

service	A namespace prefix grouping tokens for one API. Default "apifetch".

Value

A character vector of token names, empty if none are found.

Examples

```
af_list_tokens(service = "BigDataPE")
```

---

af_paginate *Pagination strategies*

Description

Constructors describing how limit/offset are sent with a request. Pass the result to the pagination argument of af_api().

Usage

```
af_paginate_offset(
  where = c("header", "query"),
  limit_param = "limit",
  offset_param = "offset"
)

af_paginate_none()
```

Arguments

where	Either "header" or "query".
limit_param, offset_param	Parameter names to use.

Details

- af_paginate_offset(): send limit/offset either as HTTP headers (default, as the Big Data PE API expects) or as URL query parameters.
- af_paginate_none(): send no pagination parameters.

Non-positive or infinite values are omitted from the request.

Value

An apifetch_pagination object.

Examples

```
af_paginate_offset("header")
af_paginate_offset("query", limit_param = "per_page")
af_paginate_none()
```

`af_remove_token` *Remove a stored API token*

Description

Removes the token stored for `name` under `service`. Does nothing (beyond a warning) when no token is found.

Usage

```
af_remove_token(name, service = "apifetch")
```

Arguments

`name`	The identifier for this token (e.g. a dataset or resource name).
`service`	A namespace prefix grouping tokens for one API. Default `"apifetch"`.

Value

Invisibly `NULL`; called for its side effect.

Examples

```
af_remove_token("dengue", service = "BigDataPE")
```

`af_store_token` *Store an API token in an environment variable*

Description

Stores an authentication token in a process environment variable named `"<service>_<name>"`. The token is never written to disk. If a non-empty variable with that name already exists, the function refuses to overwrite it.

Usage

```
af_store_token(name, token, service = "apifetch")
```

Arguments

name	The identifier for this token (e.g. a dataset or resource name).
token	The authentication token (character).
service	A namespace prefix grouping tokens for one API. Default "apifetch".

Value

Invisibly NULL; called for its side effect.

Examples

```
bdpe <- af_store_token("dengue", "your-token-here", service = "BigDataPE")
```

---

parse_queries *Build a URL with query parameters*

Description

Appends a named list of query parameters to a base URL, URL-encoding both names and values and dropping parameters whose value is the empty string.

Usage

```
parse_queries(url, query_list)
```

Arguments

url	The base URL.
query_list	A named list of query parameters.

Value

The URL with the query string appended (or the base URL unchanged when there are no parameters to add).

Examples

```
parse_queries("https://example.com", list(a = "1", b = "2"))
```

Index