

Package: vespadb (via r-universe)

October 9, 2026

Title Read Data from the vespa-db Database

Version 0.1.0.9000

Description Read data from the vespa-db database hosted at
<https://nesten.vespawatch.be>.

License MIT + file LICENSE

Depends R (>= 4.1.0)

Imports askpass, assertthat, cli, dplyr, glue, httr2, jsonlite,
magrittr, purrr, readr, rlang

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

VignetteBuilder knitr

Config/pak/sysreqs libssl-dev libx11-dev

Repository <https://inbo.r-universe.dev>

Date/Publication 2026-08-11 14:09:30 UTC

RemoteUrl <https://github.com/inbo/vespadb>

RemoteRef HEAD

RemoteSha 4a0e2a4ab550c63728112c5b75d7ddd265a19785

Contents

fetch_result	2
get_field_definitions	3
get_observation_parameters	3
get_params	4
get_vespadb_export	5
get_vespadb_obs	6
login_vespadb	7
municipalities	8

fetch_result	<i>Fetches the result from a given URL with optional query parameters and authentication.</i>
--------------	---

Description

Fetches the result from a given URL with optional query parameters and authentication.

Usage

```
fetch_result(url, .query_params = list(), .auth = NULL, return_request = FALSE)
```

Arguments

url	The API endpoint to fetch the result from.
.query_params	Optional query parameters to include in the request.
.auth	Optional path to authentication cookie to use.
return_request	If TRUE, the function returns the request object instead of the response.

Value

The response object or the request object if return_request is TRUE.

Examples

```
## Not run:
# Fetch result from a URL
fetch_result("https://uat-db.vespawatch.be/observations/30608")

# Fetch result with query parameters
fetch_result("https://uat-db.vespawatch.be/observations/",
             .query_params = list(municipality_id = 215))
# Return request object instead of response
fetch_result("https://uat-db.vespawatch.be/observations/30608",
             return_request = TRUE)

## End(Not run)
```

`get_field_definitions` *Get field definitions*

Description

This function retrieves the field definitions for the vespa-db Observations API schema. This is an overview of the information we have for every record. Not all columns that are in this table are returned in the API endpoints.

Usage

```
get_field_definitions(domain = c("production", "uat"), check_required = TRUE)
```

Arguments

<code>domain</code>	The domain to query. Either "uat" for the testing instance, or "prod" for the production instance.
<code>check_required</code>	Logical indicating whether to check if any fields are required. If no required fields are in the definition a warning is returned.

Value

A tibble with the field definitions.

Examples

```
## Not run:  
# complete field definitions  
get_field_definitions("production")  
  
## End(Not run)
```

`get_observation_parameters`

*Show an overview of the available parameters for
[get_vespadb_obs\(\)](#)*

Description

This function retrieves the parameters that can be used in the [get_vespadb_obs\(\)](#) function. It will return a tibble with the parameters that can be used in the query.

Usage

```
get_observation_parameters(domain = c("production", "uat"), ...)
```

Arguments

domain The domain to login to. Default is "uat" for the UAT environment.
 ... Additional parameters to internal methods.

Value

A tibble with the parameters that can be used in the `get_vespadb_obs()` function.

Examples

```
get_observation_parameters()
```

get_params

Get parameters for a given path in the vespa-db openAPI docs

Description

This function will return you the allowed parameters for a given API call. By default it will return all fields allowed in a `get_vespadb_obs()` query.

Usage

```
get_params(  
  path = "observations",  
  request_type = "get",  
  domain = c("production", "uat")  
)
```

Arguments

path The endpoint to get the parameters for, only GET endpoints are supported.
 request_type The type of request to get the parameters for, either "get" or "post".
 domain The domain to login to. Default is "uat" for the UAT environment.

Value

A tibble with the parameters for the given path.

Examples

```
get_params()
```

get_vespadb_export	<i>get_vespadb_export</i>
--------------------	---------------------------

Description

This function returns all observations from the VespaDB API export.

Usage

```
get_vespadb_export(domain = c("production", "uat"))
```

Arguments

domain	The domain to query. Either "uat" for the testing instance, or "prod" for the production instance.
--------	--

Details

Daily around 4 AM a export is created from the VespaDB API and stored in S3. This function retrieves the latest export from S3 and returns it as a tibble.

This function is useful for getting a snapshot of the data in the VespaDB API or for getting a large amount of data.

Value

A tibble with the observations from the VespaDB API export.

Examples

```
## Not run:  
# Get the latest export from the VespaDB API  
get_vespadb_export()  
# Get the latest export from the VespaDB API in the production instance  
get_vespadb_export(domain = "production")  
  
## End(Not run)
```

get_vespadb_obs	<i>get_vespadb_obs</i>
-----------------	------------------------

Description

Query the vespa-db api to get observations

Usage

```
get_vespadb_obs(..., auth = NULL, domain = c("production", "uat"))
```

Arguments

...	<dynamic-dots> Query parameters according to the observation model.
auth	Character. Path to the authentication cookie to use. By default a request without authentication is placed (public data only). The authentication cookie can be created using login_vespadb() which returns the path to the cookie.
domain	Character. The domain to login to. Default is "uat" for the UAT environment.

Details

Extra query parameters are supported according to the Observation model. More information on the allowed parameters can be fetched via [get_field_definitions\(\)](#).

This function already handles paging, so supplying a page as a parameter is not allowed. For a more low level function to fetch results see [fetch_result\(\)](#).

Value

A tibble with the observations from the VespaDB API.

Examples

```
## Not run:
# All public records
get_vespadb_obs()

# Public records from Ghent
get_vespadb_obs(municipality_id = 215)

# Public and private information from Ghent, only ANB plots
get_vespadb_obs(
  municipality_id = 215,
  anb = "true",
  auth = login_vespadb()
)

## End(Not run)
```

login_vespadb	<i>Login to VespaDB API to fetch fields that are not publicly available</i>
---------------	---

Description

This function allows you to login to the API and returns the required csrftoken needed to fetch information that is not publicly available.

Usage

```
login_vespadb(username, password, domain = c("production", "uat"))
```

Arguments

username	The username to login with, if not provided it will be read from the environment variable VESPADB_USER. If that is not set, you will be prompted to enter their username.
password	The password to login with, if not provided it will be read from the environment variable VESPADB_PWD. If that is not set, you will be prompted to enter their password.
domain	The domain to login to. Default is "production" for the UAT environment.

Value

The path to an authentication cookie.

Examples

```
## Not run:
get_vespadb_obs(
  municipality = 215,
  auth = login_vespadb(
    "username",
    "password"
  )
)

## End(Not run)
```

municipalities	<i>Get the municipality ids.</i>
----------------	----------------------------------

Description

This function either returns a complete overview of all municipality ids. Or if one or more municipalities are passed, the id's for these municipalities.

Usage

```
municipalities(municipalities = NULL, domain = c("production", "uat"))
```

Arguments

municipalities (Optional) A character vector of municipality names. If this is not provided, all municipalities will be returned as a tibble with their ids.

domain Character. The domain to login to. Default is "uat" for the UAT environment.

Value

Either a numeric vector of ids, or if no argument is passed, a tibble with all municipalities and their id.

Examples

```
municipalities("Gent")  
municipalities(c("Gent", "Kortrijk"))  
municipalities()
```

Index

`fetch_result`, [2](#)
`fetch_result()`, [6](#)

`get_field_definitions`, [3](#)
`get_field_definitions()`, [6](#)
`get_observation_parameters`, [3](#)
`get_params`, [4](#)
`get_vespadb_export`, [5](#)
`get_vespadb_obs`, [6](#)
`get_vespadb_obs()`, [3](#), [4](#)

`login_vespadb`, [7](#)
`login_vespadb()`, [6](#)

`municipalities`, [8](#)