Package 'deltabreedquery'

November 13, 2026

Type Package

Title Fast, Simple API Tools for Retrieving Data from 'DeltaBreed'

Version 1.0.5

Maintainer Tyr Wiesner-Hanks <twiesnerhanks@ufl.edu>

Description Simplified data retrieval from the 'DeltaBreed' breeding data management platform (<https://sandbox.breedinginsight.net/>) via the 'BrAPI' open-source breeding data API (<https://brapi.org/specification>). Each of the four main data types stored in 'DeltaBreed' (germplasm, trait variables, experiments/environments, and observations) are handled by a get_datatype() function that constructs and executes the request, handles paginated responses, and parses the retrieved data. Responses are reformatted into a consistent, R-friendly data frame format that resembles how the data appears on the 'DeltaBreed' web interface as closely as possible.

License Apache License 2.0

Encoding UTF-8

Suggests testthat (>= 3.0.0), vcr

Config/testthat/edition 3

Depends R (>= 4.1)

Imports httr2, dplyr, tidyr, rlang, jsonlite

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Tyr Wiesner-Hanks [aut, cre] (ORCID: <https://orcid.org/0000-0002-3280-5749>)

Repository CRAN

Date/Publication 2026-11-13 14:10:02 UTC

Contents

check_auth 2
filter_observations 2
get_experiments 4

get_germplasm 5
get_observations 5
get_variables 6
login_deltabreed 7
logout_deltabreed 8

Index **9**

`check_auth` *Validate BrAPI authentication credentials*

Description

Checks if the user has credentials currently stored and validates them by performing a test call to the BrAPI endpoint. Also prints the remaining time until the access token expires, if applicable.

Usage

```
check_auth()
```

Value

No return value, called for side effects (printing status)

Examples

```
check_auth()
login_deltabreed("example", verbose = FALSE)
check_auth()
```

`filter_observations` *Retrieve a filtered list of observation data*

Description

This function retrieves a filtered list of observation data using one or more filter values provided by the user, packaging the response into a data frame of the same format as `get_observations()`.

This function is best used by calling `get_experiments()` first, in order to see what experiments are actually present in the target DeltaBreed instance.

Usage

```
filter_observations(
  year = NULL,
  location = NULL,
  exp_name = NULL,
  env_name = NULL,
  exp_type = NULL,
  page_size = 1e+05,
  drop_empty_columns = FALSE,
  include_dbids = FALSE,
  case_correct = TRUE,
  verbose = TRUE
)
```

Arguments

year	A year or vector of years.
location	A location name or vector of locations.
exp_name	An experiment name or vector of names.
env_name	An environment name or vector of names.
exp_type	An experiment type or vector of types.
page_size	Page size to use for the response. Larger page sizes may decrease total retrieval time.
drop_empty_columns	If TRUE, all empty columns (including metadata columns) will be dropped from the retrieved data frame.
include_dbids	If TRUE, the alphanumeric DBIDs used by DeltaBreed will be included in the retrieved data frame.
case_correct	If FALSE, any categorical variables with case mismatches to the trait variable definition (e.g. a value of "Blue" in the observation data, when the defined trait categories are "red" and "blue") will be masked as NA instead of being case-corrected to the appropriate values.
verbose	If FALSE, all non-essential messages will be suppressed.

Value

A data frame of observations using the supplied filters.

Examples

```
login_deltabreed("example", verbose = FALSE)

# the available filters (and values to filter on) correspond to columns of get_experiments()
get_experiments()

all_obs <- get_observations(verbose = FALSE)
manitoba <- filter_observations(location = "Manitoba")
```

```
table(all_obs$Location)
table(manitoba$Location)

# all filters can take vectors as arguments
almtis <- filter_observations(location = c("Alma", "Tisdale"))
table(almtis$Year, almtis$Location)

# filters can be combined as needed
ayt_mb <- filter_observations(exp_type = "AYT", location = "Manitoba")
table(ayt_mb$ExpName, ayt_mb$EnvName)
```

`get_experiments` *Retrieve experiment summary*

Description

Retrieves a summary of all experiments and environments in a given DeltaBreed program. This may include experiments for which no observations have been recorded yet.

Usage

```
get_experiments(verbose = TRUE, include_dbids = FALSE)
```

Arguments

`verbose`	If `FALSE`, all non-essential messages will be suppressed.
`include_dbids`	If `TRUE`, the alphanumeric DBIDs used by DeltaBreed for each experiment/environment will be included in the retrieved data frame. Typically used for debugging or merging data from other sources.

Value

Data frame of experiment/environment metadata.

Examples

```
login_deltabreed("example", verbose = FALSE)

expt <- get_experiments()
expt
```

get_germplasm *Retrieve germplasm records*

Description

Retrieves all germplasm data from the current DeltaBreed instance, reformatting it to match DeltaBreed layout.

Usage

```
get_germplasm(page_size = 10000)
```

Arguments

page_size	Page size to use for the response. Larger page sizes may decrease total retrieval time.

Value

Data frame of germplasm/accession/entry information.

Examples

```
login_deltabreed("example", verbose = FALSE)

germplasm <- get_germplasm()
head(germplasm)
```

get_observations *Retrieve all observation data*

Description

Retrieves all observation units and observation data from a DeltaBreed instance via BrAPI call, converting it into a data frame resembling the data tables on the Experiments & Observations tab of DeltaBreed.

Usage

```
get_observations(
  page_size = 1e+05,
  drop_empty_columns = FALSE,
  include_dbids = FALSE,
  case_correct = TRUE,
  verbose = TRUE
)
```

Arguments

page_size	Page size to use for the response. Larger page sizes may decrease total retrieval time.
drop_empty_columns	If TRUE, all empty columns (including metadata columns) will be dropped from the retrieved data frame.
include_dbids	If TRUE, the alphanumeric DBIDs used by DeltaBreed will be included in the retrieved data frame.
case_correct	If FALSE, any categorical variables with case mismatches to the trait variable definition (e.g. a value of "Blue" in the observation data, when the defined trait categories are "red" and "blue") will be masked as NA instead of being case-corrected to the appropriate values.
verbose	If FALSE, all non-essential messages will be suppressed.

Value

A data frame of all observation units and any observations (phenotypes).

Examples

```
login_deltabreed("example", verbose = FALSE)
obs <- get_observations()

# phenotype columns will be typed according to the observation variables (trait definitions)
str(obs[,16:20])
get_variables()
```

---

get_variables *Retrieve observation variables (trait definitions) from a DeltaBreed instance.*

Description

Retrieves trait data from a DeltaBreed program via BrAPI, converting it into a data frame that mimics the appearance of the Ontology table in DeltaBreed itself.

Usage

```
get_variables(verbose = TRUE, include_archived = FALSE)
```

Arguments

verbose	If FALSE, all non-essential messages will be suppressed.
include_archived	If TRUE, archived (non-active) variables will be included in the retrieved data frame.

Value

Data frame of trait definitions drawn from BrAPI /variables endpoint.

Examples

```
login_deltabreed("example", verbose = FALSE)

vars <- get_variables()
vars
```

`login_deltabreed` *Log in to a DeltaBreed instance*

Description

This function stores your authentication credentials for a target DeltaBreed instance. To log in, you will require the BrAPI Base URL and a valid authentication token, both of which can be found on the 'BrAPI' tab of DeltaBreed.

The URL and token can be given as arguments or supplied to function prompts. The function performs some basic checks, including verifying that the user has internet access and making some test calls to the BrAPI server.

Usage

```
login_deltabreed(base_url = NULL, access_token = NULL, verbose = TRUE)
```

Arguments

`base_url`	The BrAPI Base URL, found on the BrAPI tab of DeltaBreed in the BrAPI Information pane.
`access_token`	A valid Access Token, retrieved from the DeltaBreed user interface.
`verbose`	If `FALSE`, all non-essential messages will be suppressed.

Details

Access tokens are valid for 24 hours after generation. To check your authorization credentials at any time, use the `check_auth()` function.

Value

No return value, called for side effects (storing credentials)

Examples

```
## Not run:
# function can be run with no arguments to bring up a prompt to enter the URL/token
login_deltabreed()

# since your program's URL will remain static, you can supply it as an argument, e.g.:
login_deltabreed("https://app.breedinginsight.net/v1/programs/f152169d-049f-4a7c-b5d8-c725b14e66f0")

## End(Not run)
```

`logout_deltabreed` *Clear DeltaBreed authentication credentials*

Description

Removes stored credentials (URL and access token) from the package environment. The access token will remain valid for as long as the DeltaBreed instance specifies, but in order to retrieve data the URL/token will need to be-entered with `login_deltabreed()`.

Usage

```
logout_deltabreed()
```

Value

No return value, called for side effects (clearing credentials)

Index

check_auth, 2
check_auth(), *7*

filter_observations, 2

get_experiments, 4
get_experiments(), *2*
get_germplasm, 5
get_observations, 5
get_observations(), *2*
get_variables, 6

login_deltabreed, 7
login_deltabreed(), *8*
logout_deltabreed, 8