Package ‘readepi’

October 1, 2025

Title Read Data from Relational Database Management Systems and Health Information Systems

Version 1.0.1

Description Import Data from Relational Database Management Systems (RDBMS) and Health Information Systems ('HIS'). The current version of the package supports importing data from RDBMS including 'MS SQL', 'MySQL', 'PostGRESQL', and 'SQLite', as well as from two HIS platforms: 'DHIS2' and 'SORMAS'.

License MIT + file LICENSE

URL https://epiverse-trace.github.io/readepi/,
https://github.com/epiverse-trace/readepi/,
https://github.com/epiverse-trace/readepi

BugReports https://github.com/epiverse-trace/readepi/issues

Depends R (>= 4.1.0)

Imports checkmate, cli, DBI, dplyr, httr2, odbc, pool, purrr, RMySQL, RSQLite, tidyr

Suggests cyclocomp, DiagrammeR, httptest, kableExtra, knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/Needs/website epiverse-trace/epiversetheme

Config/testthat/edition 3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

SystemRequirements odbc, libmariadbclient-dev

NeedsCompilation no

Author Karim Mané [aut, cre, cph] (ORCID: <https://orcid.org/0000-0002-9892-2999>),
Emmanuel Kabuga [ctb] (ORCID: <https://orcid.org/0009-0006-2291-709X>),

Bubacarr Bah [ctb] (ORCID: <https://orcid.org/0000-0003-3318-6668>),
Bankolé Ahadzie [ctb],
Nuredin Mohammed [ctb],
Abdoelnaser Degoot [ctb],
Hugo Gruson [rev] (ORCID: <https://orcid.org/0000-0002-4094-1476>),
Pratik Gupte [rev] (ORCID: <https://orcid.org/0000-0001-5294-7819>),
Andree Valle-Campos [rev] (ORCID: <https://orcid.org/0000-0002-7779-481X>),
London School of Hygiene and Tropical Medicine, LSHTM [cph],
data.org [fnd]

Maintainer Karim Mané <karim.mane@lshtm.ac.uk>

Repository CRAN

Date/Publication 2025-10-01 07:00:02 UTC

Contents

`dhis2_login` *Establish connection to a DHIS2 instance*

Description

Establish connection to a DHIS2 instance

Usage

```
dhis2_login(base_url, user_name, password)
```

Arguments

`base_url`	A character with the base URL of the target DHIS2 instance
`user_name`	A character with the user name
`password`	A character with the user's password

Value

An `httr2_response` object if the connection was successfully established

Examples

```
## Not run:
  dhis2_log <- dhis2_login(
    base_url = "https://play.im.dhis2.org/stable-2-42-1",
    user_name = "admin",
    password = "district"
  )

## End(Not run)
```

`get_api_version` *Get DHIS2 API version*

Description

Get DHIS2 API version

Usage

```
get_api_version(login)
```

Arguments

`login`	A httr2_response object returned by the `dhis2_login()` function

Value

A numeric with minor version of the API

Examples

```
## Not run:
  # login to the DHIS2 instance
  dhis2_login <- login(
    type = "dhis2",
    from = "https://smc.moh.gm/dhis",
    user_name = "test",
    password = "Gambia@123"
  )

  # get the API version
  api_version <- get_api_version(login = dhis2_login)

## End(Not run)
```

get_data_elements *Get all data elements from a specific DHIS2 instance*

Description

Get all data elements from a specific DHIS2 instance

Usage

```
get_data_elements(login)
```

Arguments

`login`	A httr2_response object returned by the `dhis2_login()` function

Value

A data frame with the following two columns: the data elements IDs and their corresponding names.

Examples

```
## Not run:
  # establish the connection to the system
  dhis2_login <- login(
    type = "dhis2",
    from = "https://smc.moh.gm/dhis",
    user_name = "test",
    password = "Gambia@123"
  )

  # retrieve the data elements
  data_elements <- get_data_elements(login = dhis2_login)

## End(Not run)
```

get_organisation_units

Get the organization units from a specific DHIS2 instance

Description

Retrieves all organisational reporting units and their levels, then builds a hierarchy for each unit by tracing its ancestries from the deepest level up to the root.

Usage

```
get_organisation_units(login)
```

Arguments

login	A httr2_response object returned by the `dhis2_login()` function

Details

1. Fetches all organisation units via the `get_org_units()` function,
2. Fetches all organisational unit levels via the `get_org_unit_levels()` function,
3. Filters for organisational units at the deepest level,
4. Traces the parent hierarchy of each deepest unit up to the root,
5. Constructs a tabular structure where each row is a full lineage.

Value

A data frame where each row represents a full hierarchy for the last-level unit by keeping the hierarchical organizational unit's name and ID at each level, using the official level names provided by the DHIS2 instance like "Country Name", "Country ID", etc.

Examples

```
## Not run:
  # establish the connection to the system
  dhis2_login <- login(
    type = "dhis2",
    from = "https://smc.moh.gm/dhis",
    user_name = "test",
    password = "Gambia@123"
  )

  # fetch the organisation units
  org_units <- get_organisation_units(login = dhis2_login)

## End(Not run)
```

get_programs *get all programs from a given specific DHIS2 instance*

Description

The function first fetches all programs from the DHIS2 Aggregate system, then distinguishes the Tracker and Aggregate programs.

Usage

```
get_programs(login)
```

Arguments

login	A httr2_response object returned by the dhis2_login() function

Value

A data frame with the following columns: the program ID, the program name, and the program type specifying whether the program is part of the Aggregate or Tracker system.

Examples

```
## Not run:
  # establish the connection to the system
  dhis2_login <- login(
    type = "dhis2",
    from = "https://smc.moh.gm/dhis",
    user_name = "test",
    password = "Gambia@123"
  )

  # fetch the programs
  programs <- get_programs(login = dhis2_login)

## End(Not run)
```

get_program_org_units *Get organisation units that are associated with a given program*

Description

Get organisation units that are associated with a given program

Usage

```
get_program_org_units(login, program, org_units = NULL)
```

Arguments

`login` A httr2_response object returned by the `dhis2_login()` function

`program` A character with the program ID or name

`org_units` A data frame with all organisation units from target DHIS2 instance. This is the output from the `get_organisation_units()` function

Value

A data frame with the organisation units associated with the provided program

Examples

```
## Not run:
  # login to the DHIS2 instance
  dhis2_login <- login(
    type = "dhis2",
    from = "https://smc.moh.gm/dhis",
    user_name = "test",
    password = "Gambia@123"
  )

  # fetch the organisation units
  org_units <- get_organisation_units(login = dhis2_login)

  # get the organisation units associated with the following program
  'E5IUQuHg3Mg'
  target_org_units <- get_program_org_units(
    login = dhis2_login,
    program = "E5IUQuHg3Mg",
    org_units = org_units
  )

## End(Not run)
```

`get_program_stages` *Get program stages for one or more DHSI2 programs*

Description

Retrieves the stages associated with specified DHIS2 program IDs, or all programs if none are specified. If any of the supplied program names or IDs are not found, the function displays a message and proceeds with the valid ones.

Usage

```
get_program_stages(login, programs = NULL, program = NULL)
```

Arguments

`login` A httr2_response object returned by the `dhis2_login()` function

`programs` A data frame with the program IDs and names obtained from the `get_programs()` function

`program` A character with the program ID or name

Value

A data frame with the following columns:

1. `program_id`: the unique ID of the program
2. `program_name`: the displayed name of the program
3. `program_stage_name`: the name of each stage associate with the program
4. `program_stage_id`: the ID of each program stage

Examples

```
## Not run:
  # establish the connection to the DHIS2 instance
  dhis2_login <- login(
    type = "dhis2",
    from = "https://smc.moh.gm/dhis",
    user_name = "test",
    password = "Gambia@123"
  )

  # get the list of all program stages from the DHIS2 instance
  all_program_stages <- get_program_stages(
    login = dhis2_login,
    program = "E5IUQuHg3Mg",
    programs = NULL
  )

## End(Not run)
```

`get_tracked_entities` *Get tracked entity attributes, their corresponding IDs and event IDs*

Description

Get tracked entity attributes, their corresponding IDs and event IDs

Usage

```
get_tracked_entities(login, api_version, org_unit, program, org_units)
```

Arguments

`login` A httr2_response object returned by the `dhis2_login()` function

`api_version` A numeric with the API version obtained from the `get_api_version()` function

`org_unit` A character with the organisation unit ID or name

`program` A character with the program ID or name

`org_units` A data frame with all organisation units from target DHIS2 instance. This is the output from the `get_organisation_units()` function

Value

A data frame with the tracked entity attributes alongside their events and tracked entity IDs

Examples

```
## Not run:
  # login to the DHIS2 instance
  dhis2_login <- login(
    type = "dhis2",
    from = "https://smc.moh.gm/dhis",
    user_name = "test",
    password = "Gambia@123"
  )

  # set the program and org unit IDs
  program <- "E5IUQuHg3Mg"
  org_unit <- "GcLhRNAFppR"

  # get the api version
  api_version <- get_api_version(login = dhis2_login)

  # get all the organisation units from the DHIS2 instance
  org_units <- get_organisation_units(login = dhis2_login)

  # get the tracked entity attributes
  tracked_entity_attributes <- get_tracked_entities(
    login = dhis2_login,
    api_version = api_version,
    org_unit = org_unit,
    program = program,
    org_units = org_units
  )

## End(Not run)
```

login *Establish a connection to the HIS of interest.*

Description

The current version of the package supports basic authentication (using the username and password) and personal authentication key (using API key and bearer token).

Usage

```
login(
  from,
  type,
  user_name = NULL,
  password = NULL,
  driver_name = NULL,
  db_name = NULL,
  port = NULL
)
```

Arguments

`from`	The URL to the HIS of interest. For APIs, this must be the base URL (required).
`type`	The source name (required). The current version of the package covers the following RDBMS and HIS types: "ms sql", "mysql", "postgresql", "sqlite", "dhis2", and "sormas".
`user_name`	The user name (optional).
`password`	The user's password (optional). When the password is not provided (set to `NULL`), the user will be prompt to enter the password.
`driver_name`	The driver name (optional). This is only needed for connecting to RDBMS only.
`db_name`	The database name (optional). This is only needed for connecting to RDBMS only.
`port`	The port ID (optional). This is only needed for connecting to RDBMS only.

Value

A connection object

Examples

```
# connect to the test MySQL server
## Not run:
  login <- login(
    from = "mysql-rfam-public.ebi.ac.uk",
    type = "mysql",
    user_name = "rfamro",
      password = "",
      driver_name = "",
      db_name = "Rfam",
      port = 4497
    )

## End(Not run)

# connect to a DHIS2 instance
## Not run:
  dhi2s_login <- login(
    type = "dhis2",
    from = "https://smc.moh.gm/dhis",
    user_name = "test",
    password = "Gambia@123"
  )

## End(Not run)

# connect to SORMAS
## Not run:
  sormas_login <- login(
    type = "sormas",
    from = "https://demo.sormas.org/sormas-rest",
    user_name = "SurvSup",
    password = "Lk5R7JXeZSEc"
  )

## End(Not run)
```

lookup_table *Lookup table*

Description

The lookup table is used to establish the correspondence between R operators, provided in an expression, and the DHIS2 or SQL operators.

Usage

```
lookup_table
```

Format

A data frame of 11 rows and 3 columns

read_dhis2 *Import data from DHIS2*

Description

Import data from DHIS2

Usage

```
read_dhis2(login, org_unit, program)
```

Arguments

login	A httr2_response object returned by the `dhis2_login()` function
org_unit	A character with the organisation unit ID or name
program	A character with the program ID or name

Value

A data frame that contains both the tracked entity attributes and their event data.

Examples

```
## Not run:
  # login to the DHIS2 instance
  dhis2_login <- login(
    type = "dhis2",
    from = "https://smc.moh.gm/dhis",
    user_name = "test",
    password = "Gambia@123"
  )
  program = "E5IUQuHg3Mg"
  org_unit = "GcLhRNAFppR"
  data <- read_dhis2(
    login = dhis2_login,
    org_unit = org_unit,
    program = program
  )

  # fetch data from the test DHIS2 instance
  dhis2_login <- login(
    type = "dhis2",
    from = "https://play.im.dhis2.org/stable-2-42-1",
    user_name = "admin",
    password = "district"
  )
  org_unit <- "DiszpKrYNg8"
  program <- "IpHINAT79UW"
```

```
    data <- read_dhis2(
      login = dhis2_login,
      org_unit = org_unit,
      program = program
    )

## End(Not run)
```

read_rdbms *Import data from relational database management systems (RDBMS).*

Description

The function assumes the user has read access to the database. Importing data from RDBMS requires the installation of the appropriate `driver` that is compatible with the server version hosting the database. For more details, see the vignette on how to install the driver.

Usage

```
read_rdbms(login, query)
```

Arguments

`login`	The connection object obtained from the `login()` function.
`query`	An SQL query or a list with the following elements: 1. table: a string with the table name 2. fields: a vector of column names. When specified, only those columns will be returned. Default is `NULL`. 3. filter: an expression or a vector of values used to filter the rows from the table of interest. This should be of the same length as the value for the 'select'. Default is `NULL`.

Value

A `data.frame` with the requested data as specified in the `query` argument.

Examples

```
## Not run:
  # establish the connection to the database
  rdbms_login <- login(
    from = "mysql-rfam-public.ebi.ac.uk",
    type = "mysql",
    user_name = "rfamro",
    password = "",
    driver_name = "",
    db_name = "Rfam",
    port = 4497
  )

  # import data where query parameters are specified as a list
  authors_list <- read_rdbms(
    login = rdbms_login,
    query = list(table = "author", fields = NULL, filter = NULL)
  )

  # import data where query parameters is within an SQL query
  authors_list <- read_rdbms(
    login = rdbms_login,
    query = "select * from author"
  )

## End(Not run)
```

read_sormas *Import data from SORMAS*

Description

The function returns the following columns by default: case_id,person_id, sex, date_of_birth, case_origin, country, city, latitude,longitude, case_status, date_onset, date_admission, outcome, date_outcome,contact_id, date_last_contact, date_first_contact, Ct_values.

Usage

```
read_sormas(login, disease, since = 0)
```

Arguments

login	A list with the user's authentication details
disease	A character with the target disease name
since	A Date value in ISO8601 format (YYYY-mm-dd). Default is 0 i.e. to fetch all cases from the beginning of data collection.

Details

Note that the some values in the date_of_birth column of the output object might not have some missing elements such a missing year (NA-12-26), month (2025-NA-01) or date (2025-12-NA), or a combination of two missing elements.

Value

A data frame with the case data of the specified disease.

Examples

```
## Not run:
  # establish the connection to the SORMAS system
  sormas_login <- login(
    type = "sormas",
    from = "https://demo.sormas.org/sormas-rest",
    user_name = "SurvSup",
    password = "Lk5R7JXeZSEc"
  )
  # fetch all COVID (coronavirus) cases from the test SORMAS instance
  covid_cases <- read_sormas(
    login = sormas_login,
    disease = "coronavirus"
  )

## End(Not run)
```

request_parameters *Request parameters*

Description

The request_parameters data frame contains the information related to which request parameters should be used when fetching data from a specific version of DHIS2. It currently contains parameters for versions ranging from 2.22 to 2.42.

Usage

```
request_parameters
```

Format

A data frame of 3 rows and 12 columns

show_tables *Display the list of tables in a database*

Description

Display the list of tables in a database

Usage

```
show_tables(login)
```

Arguments

`login`	The connection object obtained from the `login()` function.

Value

a `character` that contains the list of all tables found in the specified database.

Examples

```
## Not run:
# connect to the test MySQL server
  rdbms_login <- login(
    from        = "mysql-rfam-public.ebi.ac.uk",
    type        = "mysql",
    user_name   = "rfamro",
    password    = "",
    driver_name = "",
    db_name     = "Rfam",
    port        = 4497
  )

# display the list of available tables from this database
tables <- show_tables(login = rdbms_login)

## End(Not run)
```

sormas_get_api_specification

Download the API specification file from SORMAS

Description

Download the API specification file from SORMAS

Usage

```
sormas_get_api_specification(path = tempdir(), overwrite = TRUE)
```

Arguments

`path`	A character with the path to the folder where the downloaded data dictionary should be stored. Default is `NULL` i.e. the data dictionary will be stored in `tempdir()`
`overwrite`	A logical used to specify whether to overwrite to overwrite the existing data dictionary or not. Default is `TRUE`

Value

Invisibly returns the path to the folder where the file is stored. When `path = NULL`, the file will be stored in the R temporary folder as: `api_specification.json`

Examples

```
# save the SORMAS API specification into the temporary R folder
path_api_specs <- sormas_get_api_specification()
```

sormas_get_data_dictionary

Download SORMAS data dictionary

Description

Download SORMAS data dictionary

Usage

```
sormas_get_data_dictionary(path = tempdir(), overwrite = TRUE)
```

Arguments

`path`	A character with the path to the folder where the downloaded data dictionary should be stored. Default is `NULL` i.e. the data dictionary will be stored in `tempdir()`
`overwrite`	A logical used to specify whether to overwrite to overwrite the existing data dictionary or not. Default is `TRUE`

Value

A character with path to the folder where the data dictionary is stored. When `path = NULL`, the file will be stored in the R temporary folder as: `dictionary.xlsx`

Examples

```
# download and save the data dictionary in the default R temporary directory
path_to_dictionary <- sormas_get_data_dictionary()
```

sormas_get_diseases *Get the list of disease names from a SORMAS instance*

Description

Get the list of disease names from a SORMAS instance

Usage

```
sormas_get_diseases(login)
```

Arguments

login	A list with the user's authentication details

Value

A vector of the list of disease names in a SORMAS instance

Examples

```
## Not run:
  # establish the connection to the SORMAS system
  sormas_login <- login(
    type = "sormas",
    from = "https://demo.sormas.org/sormas-rest",
    user_name = "SurvSup",
    password = "Lk5R7JXeZSEc"
  )
  disease_names <- sormas_get_diseases(
    login = sormas_login
  )

## End(Not run)
```

Index