

Package ‘khisr’

September 16, 2026

Title An R Client to Retrieve Data from DHIS2

Description Provides a user-friendly interface for interacting with the District Health Information Software 2 ('DHIS2', <<https://dhis2.org>>) instance. It streamlines data retrieval, empowering researchers, analysts, and healthcare professionals to obtain and utilize data efficiently.

Version 1.0.8

License MIT + file LICENSE

URL <https://khisr.damurka.com>, <https://github.com/damurka/khisr>

BugReports <https://github.com/damurka/khisr/issues>

Depends R (>= 3.3)

Imports cli, curl, dplyr, httr2 (>= 1.1.0), janitor, jsonlite, lifecycle, lubridate, magrittr, purrr, rlang, stringr, tidyr, utils, withr

Suggests knitr, rmarkdown, spelling, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Language en-GB

RoxygenNote 7.3.2

NeedsCompilation no

Author David Kariuki [aut, cre, cph] (ORCID: <<https://orcid.org/0009-0003-6159-1107>>)

Maintainer David Kariuki <hello@damurka.com>

Repository CRAN

Date/Publication 2026-09-16 08:40:09 UTC

Contents

analytics-dimension	3
get_analytics	4
get_analytics_by_level	5
get_analytics_outliers	7
get_complete_data_set_registrations	8
get_data_elements_with_category_options	10
get_data_sets_by_level	11
get_data_store_keys	12
get_data_store_namespaces	13
get_data_store_value	14
get_data_value_audits	15
get_data_value_sets	16
get_enrollments	18
get_enrollment_analytics	20
get_enrollment_analytics_aggregate	21
get_events	23
get_event_analytics	25
get_event_analytics_aggregate	26
get_file_resources	28
get_geo_features	29
get_metadata	30
get_organisations_by_level	31
get_relationships	32
get_sql_views	33
get_sql_view_data	34
get_system_info	36
get_tracked_entities	36
get_validation_results	39
khisr-configuration	40
khis_api_version	41
khis_base_url	42
khis_cred	43
khis_cred_clear	45
khis_display_name	46
khis_has_cred	47
khis_username	48
metadata-filter	49
metadata-helpers	51
tracked_entity_filter	54

Index

56

Description

Constructs a dimensions expression for analytics queries based on specified property, operator, and values.

Usage

```
analytics_dimension(property, operator, values)
```

```
operator %.d% values
```

```
operator %.f% values
```

Arguments

property	A character string representing whether its dimension or filter. It only accepts 'dimension', 'filter'.
operator	A character string representing the property to filter on (e.g., 'dx', 'pe', 'ou').
values	A vector of values or semi-colon separated string items.

Details

DHIS2 organizes data using multiple dimensions, each with a unique identifier and a set of items that represent specific data points within that dimension.

- Data elements (dx): Indicators, data set reporting rate metrics, data element operands, program indicators, program data elements, program attributes, validation rules.
- Periods (pe): ISO periods (e.g., 202401) and relative periods (e.g., LAST_WEEK).
- Organisation unit hierarchy (ou): Specific health facilities, districts, countries, and keywords for user location or its sub-units.
- Category option combinations (co): Category option combo identifiers.
- Attribute option combinations (ao): Attribute option combo identifiers.
- Categories: Category option identifiers.
- Data element group sets: Data element group identifiers.
- Organisation unit group sets: Organisation unit group identifiers.

The infix operator used for filter and dimension includes:

- %.d%: Infix operator for constructing dimension filters. Equivalent to calling `analytics_dimension("dimension", ...)`.
- %.f%: Infix operator for constructing filter filters. Equivalent to calling `analytics_dimension("filter", ...)`.

Value

A spliced list with filter in the format property=operator:value

Examples

```
# Create a dimension for data element "DE_1234"
analytics_dimension('dimension', "dx", "DE_1234")

# Equivalent to the expression above
dx %.d% "DE_1234"

# Create a filter dimension for the period of January 2024
pe %.f% "202401"

# Create filter dimension for periods "202401" and "202402":
analytics_dimension("filter", "pe", c("202401", "202402"))
```

get_analytics	<i>Retrieves Disaggregated Analytics Data from DHIS2</i>
---------------	--

Description

get_analytics() retrieves disaggregated data from DHIS2 analytics tables for a specified period and data element(s), without performing any aggregation.

Usage

```
get_analytics(
  ...,
  return_type = c("uid", "name"),
  retry = 2,
  verbosity = 0,
  timeout = 60
)
```

Arguments

...	One or more <code>analytics_dimension()</code> parameters in key-value pairs. These define the analytics query, including data elements, periods, and organization units.
return_type	Optional argument specifying the return format for identifiers. defaults to 'uid'. Choose 'name' for human readable labels.
retry	Number of times to retry the API call in case of failure (defaults to 2).
verbosity	Level of HTTP information to print during the call: • 0: No output

- 1: Show headers
 - 2: Show headers and bodies
 - 3: Show headers, bodies, and CURL status message.
- timeout Maximum number of seconds to wait for the API response.

Details

- Retrieves data directly from DHIS2 analytics tables.
- Allows specifying analytics dimensions, return format for identifiers, retry attempts, and logging verbosity.

Value

A tibble containing the disaggregated analytics data, or NULL if no data is retrieved.

Examples

```
# Malaria data elements
# lYsfXxCw6Qi = MAL - Malaria confirmed cases reported
# GxlrIgMyEf4 = MAL - Malaria deaths
element_id <- c('lYsfXxCw6Qi', 'GxlrIgMyEf4')

# Download data for the last year
data <- get_analytics(dx %d% element_id, pe %d% 'LAST_YEAR')
data
```

```
get_analytics_by_level
```

Retrieves Analytics Table Data

Description

[Experimental] get_analytics_by_level() fetches data from the DHIS2 analytics tables for a given period and data element(s), without performing any aggregation.

Usage

```
get_analytics_by_level(
  element_ids,
  start_date,
  end_date = NULL,
  level = 1,
  org_ids = NULL,
  ...,
  auth = NULL,
  call = caller_env()
)
```

<code>element_ids</code>	Required vector of data element IDs for which to retrieve data.
<code>start_date</code>	Required start date to retrieve data. It is required and in the format YYYY-MM-dd.
<code>end_date</code>	Optional ending date for data retrieval (default is the current date).
<code>level</code>	The desired organisation level of data (default: level 1)
<code>org_ids</code>	Optional list of organization units IDs to be filtered.
<code>...</code>	Other analytics query options passed onto the DHIS2 analytics endpoint (e.g. additional dimension/filter arguments). Not forwarded to the organisation unit or data element metadata lookups this function also performs.
<code>auth</code>	Optional. The authentication object.
<code>call</code>	The caller environment.

- Retrieves data directly from DHIS2 analytics tables.
- Supports optional arguments for providing organization lists, data elements, and categories.
- Allows specifying DHIS2 session objects, retry attempts, and logging verbosity.

A tibble with detailed information, including:

- Geographical identifiers (country, subnational, district, facility, depending on level)
- Reporting period (month, year, fiscal year)
- Data element names
- Category options
- Reported values

- `get_organisations_by_level()` for getting the organisations units
- `get_data_elements_with_category_options()` for retrieving the data elements

[illegible]

get_analytics_outliers

Get Statistical Outliers from DHIS2 Analytics

Description

[Experimental] get_analytics_outliers() retrieves data values flagged as statistical outliers by DHIS2's analytics/outlierDetection endpoint — useful for data-quality review.

Usage

```
get_analytics_outliers(
  data_elements = NULL,
  data_sets = NULL,
  org_units,
  start_date,
  end_date,
  algorithm = NULL,
  threshold = NULL,
  ...,
  retry = 2,
  verbosity = 0,
  timeout = 60,
  auth = NULL,
  call = caller_env()
)
```

Arguments

data_elements	Optional. A vector of data element ids to check. Provide this or data_sets.
data_sets	Optional. A vector of data set ids to check. Provide this or data_elements.
org_units	A vector of organisation unit ids to scope the query to.
start_date, end_date	ISO-8601 dates bounding the query.
algorithm	Optional. The detection algorithm: 'Z_SCORE' (default), 'MOD_Z_SCORE', or 'MIN_MAX'.
threshold	Optional. The sensitivity threshold for the chosen algorithm; higher values flag fewer outliers.
...	Other query parameters supported by your DHIS2 instance's analytics/outlierDetection endpoint (e.g. maxResults, orderBy).
retry	Number of times to retry the API call in case of failure (defaults to 2).
verbosity	Level of HTTP information to print during the call.
timeout	Maximum number of seconds to wait for the API response.
auth	Optional. The authentication object.
call	The caller environment.

Details

Requires either `data_elements` or `data_sets`. **Unlike the rest of this package, this endpoint's response shape has not been independently verified live:** the public demo instance used to verify khizr's other Tracker/analytics functions returned 403 Forbidden for this endpoint regardless of parameters, which appears to be an account-authority restriction rather than a request error. The field names below follow DHIS2's published API documentation; confirm them against your own instance before relying on this function in production.

Value

A tibble of outlier values as returned by DHIS2 (typically including the data element, org unit, period, value, and statistical bounds that flagged it), or NULL if none were found.

See Also

[get_analytics\(\)](#) for the underlying aggregated values.

Examples

```
## Not run:
get_analytics_outliers(data_elements = 'lYsfXxCw6Qi',
                       org_units = 'W6sNfkJcXGC',
                       start_date = '2023-01-01',
                       end_date = '2023-12-31')

## End(Not run)
```

```
get_complete_data_set_registrations
```

Get Raw Data Set Completeness Registrations from a DHIS2 Instance

Description

[Experimental] `get_complete_data_set_registrations()` retrieves the raw completeness-registration records (who marked a data set complete, and when) from DHIS2's `completeDataSetRegistrations` endpoint — the raw counterpart to [get_data_sets_by_level\(\)](#)'s aggregated reporting-rate view, the same way [get_data_value_sets\(\)](#) complements [get_analytics\(\)](#).

Usage

```
get_complete_data_set_registrations(
  data_sets,
  org_units,
  children = NULL,
  periods,
  ...,
```

```
get_data_elements_with_category_options
```

Get Data Elements with Category Options

Description

[Experimental] `get_data_elements_with_category_options()` fetches data elements meta-data with the category options from the DHIS2 API server.

Usage

```
get_data_elements_with_category_options(  
  element_ids,  
  auth = NULL,  
  call = caller_env()  
)
```

Arguments

<code>element_ids</code>	The data element identifiers whose details being retrieved
<code>auth</code>	The authentication object
<code>call</code>	The caller environment

Value

A tibble containing the following columns:

- `element_id` - The unique identifier for the data element.
- `element` - The name of the data element.
- `category` - The category options for the elements
- `category_id` - The unique identifier for the category options

Examples

```
# Fetch the data element metadata for particular element id  
elements <- get_data_elements_with_category_options('lYsfXxCw6Qi')  
elements
```

get_data_sets_by_level

Retrieves Data Set Reporting Rate Metrics

Description

[Experimental] get_data_sets_by_level() fetches the data set reporting metrics. The metric can be REPORTING_RATE, REPORTING_RATE_ON_TIME, ACTUAL_REPORTS, ACTUAL_REPORTS_ON_TIME, EXPECTED_REPORTS.

Usage

```
get_data_sets_by_level(
  dataset_ids,
  start_date,
  end_date = NULL,
  level = 1,
  org_ids = NULL,
  ...,
  auth = NULL,
  call = caller_env()
)
```

Arguments

dataset_ids	Required vector of data sets IDs for which to retrieve data. Required.
start_date	Optional start date to retrieve data. It is required and in the format YYYY-MM-dd.
end_date	Optional ending date for data retrieval (default is the current date).
level	Required desired organisation level of data (default: level 1) .
org_ids	Optional list of organization units IDs to be filtered.
...	Other analytics query options passed onto the DHIS2 analytics endpoint (e.g. additional dimension/filter arguments). Not forwarded to the organisation unit or data set metadata lookups this function also performs.
auth	Optional. The authentication object.
call	The caller environment.

Value

A tibble with detailed information, including:

- Geographical identifiers (country, subnational, district, facility, depending on level)
- Reporting period (month, year, fiscal year)
- The reporting metric can be REPORTING_RATE, REPORTING_RATE_ON_TIME, ACTUAL_REPORTS, ACTUAL_REPORTS_ON_TIME, EXPECTED_REPORTS.

See Also

- [get_organisations_by_level\(\)](#) for getting the organisations units
- [get_data_sets\(\)](#) for retrieving the data sets

Examples

```
# The Malaria elimination dataset
dataset_id = c('VEM58nY22s0')

# Download data from February 2023 to current date
data <- get_data_sets_by_level(dataset_ids = dataset_id,
                              start_date = '2023-02-01')

data
```

get_data_store_keys	<i>Get the Keys in a DHIS2 Data Store Namespace</i>
---------------------	---

Description

[Experimental] `get_data_store_keys()` lists the keys stored under a namespace in a DHIS2 instance's key/value data store. See [get_data_store_namespaces\(\)](#).

Usage

```
get_data_store_keys(
  namespace,
  store = c("system", "user"),
  auth = NULL,
  call = caller_env()
)
```

Arguments

namespace	A data store namespace, as returned by get_data_store_namespaces() .
store	Optional. 'system' (default) or 'user'; see get_data_store_namespaces() .
auth	Optional. The authentication object.
call	The caller environment.

Value

A character vector of keys, or NULL if the namespace has none. Confirmed live against a public DHIS2 demo instance.

See Also

[get_data_store_namespaces\(\)](#), [get_data_store_value\(\)](#).

Examples

```
namespaces <- get_data_store_namespaces()
get_data_store_keys(namespaces[1])
```

```
get_data_store_namespaces
```

Get DHIS2 Data Store Namespaces

Description

[Experimental] `get_data_store_namespaces()` lists the namespaces in a DHIS2 instance's key/value data store — arbitrary JSON configuration used by DHIS2 apps (e.g. the Maps app, Tracker Capture). Use [get_data_store_keys\(\)](#) and [get_data_store_value\(\)](#) to read a namespace's contents.

Usage

```
get_data_store_namespaces(
  store = c("system", "user"),
  auth = NULL,
  call = caller_env()
)
```

Arguments

<code>store</code>	Optional. 'system' (default), the shared instance-wide store, or 'user', the store private to the authenticated user.
<code>auth</code>	Optional. The authentication object.
<code>call</code>	The caller environment.

Value

A character vector of namespaces, or NULL if none exist. Confirmed live against a public DHIS2 demo instance.

See Also

[get_data_store_keys\(\)](#), [get_data_store_value\(\)](#).

Examples

```
get_data_store_namespaces()
```

get_data_store_value *Get a Value from a DHIS2 Data Store*

Description

[Experimental] `get_data_store_value()` retrieves the JSON value stored under a namespace/key pair in a DHIS2 instance's key/value data store. See [get_data_store_namespaces\(\)](#).

Usage

```
get_data_store_value(  
  namespace,  
  key,  
  store = c("system", "user"),  
  auth = NULL,  
  call = caller_env()  
)
```

Arguments

namespace	A data store namespace, as returned by get_data_store_namespaces() .
key	A key within namespace, as returned by get_data_store_keys() .
store	Optional. 'system' (default) or 'user'; see get_data_store_namespaces() .
auth	Optional. The authentication object.
call	The caller environment.

Value

The stored value, parsed from JSON — typically a named list, but shape varies entirely by what the namespace/key holds. Confirmed live against a public DHIS2 demo instance.

See Also

[get_data_store_namespaces\(\)](#), [get_data_store_keys\(\)](#).

Examples

```
namespaces <- get_data_store_namespaces()  
keys <- get_data_store_keys(namespaces[1])  
get_data_store_value(namespaces[1], keys[1])
```

get_data_value_audits *Get Data Value Change History from a DHIS2 Instance*

Description

[Experimental] get_data_value_audits() retrieves the audit trail (who changed a data value, when, and to what) from DHIS2's audits/dataValue endpoint.

Usage

```
get_data_value_audits(
  data_elements = NULL,
  org_units = NULL,
  periods = NULL,
  ...,
  retry = 2,
  verbosity = 0,
  timeout = 60,
  auth = NULL,
  call = caller_env()
)
```

Arguments

data_elements	Optional. A vector of data element ids to scope the query to.
org_units	Optional. A vector of organisation unit ids to scope the query to.
periods	Optional. A vector of ISO period strings to scope the query to.
...	Other query parameters supported by your DHIS2 instance's audits/dataValue endpoint (e.g. categoryOptionCombo, attributeOptionCombo).
retry	Number of times to retry the API call in case of failure (defaults to 2).
verbosity	Level of HTTP information to print during the call.
timeout	Maximum number of seconds to wait for the API response.
auth	Optional. The authentication object.
call	The caller environment.

Details

Confirmed live against a public DHIS2 demo instance that this endpoint accepts these parameters and returns a valid, empty result — the data value tried had no recorded edit history, so the shape of a populated result is not independently verified.

Value

A tibble of data value audit records (data element, period, org unit, category/attribute option combos, value, modified date, audit type), or NULL if none were found.

Examples

```
get_data_value_audits(data_elements = 'lYsfXxCw6Qi',
                      org_units = 'W6sNfkJcXGC',
                      periods = '202301')
```

get_data_value_sets	<i>Get Raw Data Values from a DHIS2 Instance</i>
---------------------	--

Description

[Experimental] `get_data_value_sets()` retrieves individually entered aggregate data values from DHIS2's `dataValueSets` endpoint — as opposed to `get_analytics()`, which returns pre-aggregated, computed values from the analytics tables. Useful for data-quality auditing, or when you need the raw entered values (including who entered them and when) rather than an aggregated view.

Usage

```
get_data_value_sets(
  data_elements = NULL,
  data_sets = NULL,
  data_element_groups = NULL,
  org_units = NULL,
  org_unit_group = NULL,
  children = NULL,
  periods = NULL,
  start_date = NULL,
  end_date = NULL,
  last_updated = NULL,
  id_scheme = NULL,
  ...,
  retry = 2,
  verbosity = 0,
  timeout = 60,
  auth = NULL,
  call = caller_env()
)
```

Arguments

<code>data_elements</code>	Optional. A vector of data element ids to scope the query to.
<code>data_sets</code>	Optional. A vector of data set ids to scope the query to.
<code>data_element_groups</code>	Optional. A vector of data element group ids to scope the query to.
<code>org_units</code>	Optional. A vector of organisation unit ids to scope the query to.

org_unit_group	Optional. An organisation unit group id to scope the query to.
children	Optional. If TRUE, also includes data for the descendants of org_units, not just org_units itself. DHIS2 defaults to FALSE (exact org units only) when not set.
periods	Optional. A vector of ISO period strings (e.g. '202501'). Required unless start_date/end_date are given instead.
start_date, end_date	Optional. ISO-8601 dates bounding the query, as an alternative to periods.
last_updated	Optional. ISO-8601 date or datetime string; only returns data values updated after this point.
id_scheme	Optional. Remaps identifiers in the response (e.g. 'code') instead of the default DHIS2 UIDs.
...	Other query parameters supported by your DHIS2 instance's dataValueSets endpoint.
retry	Number of times to retry the API call in case of failure (defaults to 2).
verbosity	Level of HTTP information to print during the call.
timeout	Maximum number of seconds to wait for the DHIS2 API response.
auth	Optional. The authentication object.
call	The caller environment.

Details

DHIS2 requires this query to be scoped by at least one of `data_elements`, `data_sets`, or `data_element_groups`; by at least one of `org_units` or `org_unit_group`; and by either `periods` or `start_date/end_date`. An unscoped query will be rejected by the server. Confirmed live against a public DHIS2 demo instance: without `children = TRUE`, `org_units` matches only that exact organisation unit, not its descendants — a query that returned 0 values with `org_units` alone returned real data once `children = TRUE` was added.

Unlike the metadata and Tracker endpoints, `dataValueSets` is not paginated — it returns the full matching result set in a single response.

Value

A tibble of data values (`dataElement`, `period`, `orgUnit`, `categoryOptionCombo`, `attributeOptionCombo`, `value`, `storedBy`, `created`, `lastUpdated`, `comment`, `followup`), or NULL if none were found.

See Also

[get_analytics\(\)](#) for pre-aggregated analytics values, [get_data_sets_by_level\(\)](#) for data set reporting-rate metrics.

Examples

```
# Raw data values for a data set at an org unit and everything below it,
# for a single period
get_data_value_sets(data_sets = 'VEM58nY22s0',
```

```
org_units = 'W6sNfkJcXGC',
children = TRUE,
periods = '202401')
```

get_enrollments

Get Enrollments from a DHIS2 Instance

Description

[Experimental] get_enrollments() retrieves program enrollments from the DHIS2 Tracker API.

Usage

```
get_enrollments(
  program = NULL,
  org_units = NULL,
  org_unit_mode = NULL,
  updated_after = NULL,
  updated_before = NULL,
  enrolled_after = NULL,
  enrolled_before = NULL,
  occurred_after = NULL,
  occurred_before = NULL,
  ...,
  fields = c("enrollment", "program", "trackedEntity", "orgUnit", "status", "enrolledAt",
    "occurredAt"),
  page_size = 500,
  retry = 2,
  verbosity = 0,
  timeout = 60,
  auth = NULL,
  call = caller_env()
)
```

Arguments

program	Optional. A program id to scope the query to.
org_units	Optional. A vector of organisation unit ids to scope the query to.
org_unit_mode	Optional. One of "SELECTED", "CHILDREN", "DESCENDANTS", "ACCESSIBLE", "CAPTURE", "ALL", controlling how org_units is interpreted. DHIS2 defaults to "ACCESSIBLE" when org_units is not provided, and "SELECTED" when it is.
updated_after, updated_before	Optional. ISO-8601 date or datetime strings bounding the enrollment's last-updated timestamp.

enrolled_after, enrolled_before	Optional. ISO-8601 date or datetime strings bounding the enrollment date (sent as the unprefixed enrolledAfter/enrolledBefore query params — the corresponding get_tracked_entities() arguments use enrollmentEnrolledAfter/enrollmentEnrolledBefore instead, to disambiguate from a tracked entity's other nested dates).
occurred_after, occurred_before	Optional. ISO-8601 date or datetime strings bounding the enrollment's incident/occurred date (sent as the unprefixed occurredAfter/occurredBefore query params — the corresponding get_tracked_entities() arguments use enrollmentOccurredAfter/enrollmentOccurredBefore instead).
...	Other query parameters supported by your DHIS2 instance's Tracker API (e.g. status).
fields	The DHIS2 field-selector for the columns to return.
page_size	Number of records to request per page (default 500).
retry	Number of times to retry the API call in case of failure (defaults to 2).
verbosity	Level of HTTP information to print during the call.
timeout	Maximum number of seconds to wait for the DHIS2 API response.
auth	Optional. The authentication object.
call	The caller environment.

Details

DHIS2 requires an enrollments query to be scoped by at least one of program or org_units; an unscoped query will be rejected by the server. DHIS2's Tracker API does not document filter support for this endpoint, so a filter argument raises an error rather than being silently sent as an unsupported query parameter; use [get_tracked_entities\(\)](#) with [tracked_entity_filter\(\)](#) to filter by attribute value instead.

The enrolledAfter/enrolledBefore/occurredAfter/occurredBefore query parameter names used here (unprefixed, unlike the equivalent [get_tracked_entities\(\)](#) arguments) were confirmed by testing live against a public DHIS2 demo instance — enrolledAfter measurably narrowed the result set, and orgUnits/orgUnit both work for this endpoint (unlike events, which only accepts the singular form).

Value

A tibble of enrollments, or NULL if none were found.

See Also

[get_tracked_entities\(\)](#), [get_events\(\)](#)

Other tracker functions: [get_events\(\)](#), [get_relationships\(\)](#), [get_tracked_entities\(\)](#), [tracked_entity_filter\(\)](#)

Examples

```
# All enrollments in a program at a given org unit
get_enrollments(program = 'PREnRHSp3be',
                org_units = 'IWp9dQGM0bS',
                org_unit_mode = 'DESCENDANTS')
```

```
get_enrollment_analytics
```

Retrieves Aggregated Enrollment Analytics Data from DHIS2

Description

[Experimental] `get_enrollment_analytics()` retrieves aggregated, dimensional analytics over Tracker enrollments from DHIS2's `analytics/enrollments/query/{program}` endpoint — as opposed to `get_enrollments()`, which returns raw individual enrollment records from the Tracker API. Use this when you want counts/aggregates across dimensions (e.g. enrollments by org unit and period), not the underlying records themselves.

Usage

```
get_enrollment_analytics(
  program,
  ...,
  return_type = c("uid", "name"),
  page_size = 1000,
  retry = 2,
  verbosity = 0,
  timeout = 60,
  auth = NULL,
  call = caller_env()
)
```

Arguments

<code>program</code>	A program id. Used as a URL path segment, not a query parameter.
<code>...</code>	One or more <code>analytics_dimension()</code> parameters (e.g. <code>ou/pe</code> dimensions via <code>%d/%.f%</code>), and/or other query parameters supported by your DHIS2 instance's Tracker analytics query API.
<code>return_type</code>	Optional. <code>'uid'</code> (default) or <code>'name'</code> for the identifier scheme used in the response.
<code>page_size</code>	Number of records to request per page (default 1000).
<code>retry</code>	Number of times to retry the API call in case of failure (defaults to 2).
<code>verbosity</code>	Level of HTTP information to print during the call.

timeout	Maximum number of seconds to wait for the API response.
auth	Optional. The authentication object.
call	The caller environment.

Details

The response shares the same headers/rows shape as [get_analytics\(\)](#). Pagination is automatic; see [get_event_analytics\(\)](#)'s details for a note on this endpoint family's pagination behaviour (a different, silently truncating shape confirmed live against a public DHIS2 demo instance for the events variant, which this function shares the same underlying engine with).

Value

A tibble of aggregated enrollment analytics data, or NULL if none was retrieved.

See Also

[get_enrollments\(\)](#) for raw individual enrollment records, [get_event_analytics\(\)](#) for the event equivalent, [get_analytics\(\)](#) for aggregate (non-Tracker) analytics.

Examples

```
# Enrollments for a program, by org unit, over the last 12 months
get_enrollment_analytics(program = 'PREnRHSp3be',
  ou %.d% 'USER_ORGUNIT',
  pe %.d% 'LAST_12_MONTHS')
```

get_enrollment_analytics_aggregate

Get Aggregated (Pivot-Style) Enrollment Analytics from DHIS2

Description

[Experimental] `get_enrollment_analytics_aggregate()` retrieves totals over Tracker enrollments from DHIS2's `analytics/enrollments/aggregate/{program}` endpoint. This is a genuinely different endpoint from [get_enrollment_analytics\(\)](#)'s `analytics/enrollments/query/{program}`: query returns line-list style rows, while aggregate returns pivot-table style totals.

Usage

```
get_enrollment_analytics_aggregate(
  program,
  ...,
  return_type = c("uid", "name"),
  retry = 2,
  verbosity = 0,
```

```

    timeout = 60,
    auth = NULL,
    call = caller_env()
  )

```

Arguments

<code>program</code>	A program id. Used as a URL path segment, not a query parameter.
<code>...</code>	One or more analytics_dimension() parameters (e.g. <code>ou/pe</code> dimensions via <code>%d%/%f%</code>), and/or other query parameters supported by your DHIS2 instance's Tracker analytics aggregate API.
<code>return_type</code>	Optional. <code>'uid'</code> (default) or <code>'name'</code> for the identifier scheme used in the response.
<code>retry</code>	Number of times to retry the API call in case of failure (defaults to 2).
<code>verbosity</code>	Level of HTTP information to print during the call.
<code>timeout</code>	Maximum number of seconds to wait for the API response.
<code>auth</code>	Optional. The authentication object.
<code>call</code>	The caller environment.

Details

The response shares the same headers/rows shape as [get_analytics\(\)](#). Confirmed live to be unpaginated, like [get_event_analytics_aggregate\(\)](#).

Value

A tibble of aggregated enrollment totals, or NULL if none was retrieved.

See Also

[get_enrollment_analytics\(\)](#) for the line-list query equivalent, [get_event_analytics_aggregate\(\)](#) for the event equivalent.

Examples

```

get_enrollment_analytics_aggregate(program = 'PREnRHSp3be',
  ou = '%d%' 'USER_ORGUNIT',
  pe = '%d%' 'LAST_12_MONTHS')

```

get_events

*Get Events from a DHIS2 Instance***Description**

[Experimental] get_events() retrieves tracker program-stage events from the DHIS2 Tracker API.

Usage

```
get_events(
    program = NULL,
    program_stage = NULL,
    org_unit = NULL,
    org_unit_mode = NULL,
    updated_after = NULL,
    updated_before = NULL,
    occurred_after = NULL,
    occurred_before = NULL,
    ...,
    fields = c("event", "program", "programStage", "orgUnit", "status", "occurredAt",
               "scheduledAt", "dataValues"),
    page_size = 500,
    retry = 2,
    verbosity = 0,
    timeout = 60,
    auth = NULL,
    call = caller_env()
)
```

Arguments

program	Optional. A program id to scope the query to.
program_stage	Optional. A program stage id to scope the query to.
org_unit	Optional. A single organisation unit id to scope the query to. Unlike get_tracked_entities()/get_enr... the events endpoint accepts exactly one org unit (sent as the singular orgUnit query param, confirmed live against a DHIS2 demo instance — passing more than one is rejected by the server with a "UID must be..." error, not silently truncated).
org_unit_mode	Optional. One of "SELECTED", "CHILDREN", "DESCENDANTS", "ACCESSIBLE", "CAPTURE", "ALL", controlling how org_unit is interpreted. DHIS2 defaults to "ACCESSIBLE" when org_unit is not provided, and "SELECTED" when it is.
updated_after, updated_before	Optional. ISO-8601 date or datetime strings bounding the event's last-updated timestamp.

occurred_after, occurred_before	Optional. ISO-8601 date or datetime strings bounding the event's occurred date (sent as the unprefixd occurredAfter/occurredBefore query params — the corresponding <code>get_tracked_entities()</code> arguments use eventOccurredAfter/eventOccurredBefore instead, to disambiguate from a tracked entity's other nested dates).
...	Other query parameters supported by your DHIS2 instance's Tracker API (e.g. status, assignedUserMode).
fields	The DHIS2 field-selector for the columns to return.
page_size	Number of records to request per page (default 500).
retry	Number of times to retry the API call in case of failure (defaults to 2).
verbosity	Level of HTTP information to print during the call.
timeout	Maximum number of seconds to wait for the DHIS2 API response.
auth	Optional. The authentication object.
call	The caller environment.

Details

DHIS2 requires an events query to be scoped by at least one of program, program_stage, or org_unit; an unscoped query will be rejected by the server.

The occurredAfter/occurredBefore query parameter names used here (unprefixed, unlike the equivalent `get_tracked_entities()` arguments) and the single-value orgUnit parameter were confirmed by testing live against a public DHIS2 demo instance.

DHIS2's published Tracker API documentation does not describe a supported way to filter events by data element value (unlike `get_tracked_entities()`, which supports attribute filtering via `tracked_entity_filter()`), so a filter argument raises an error here rather than being silently sent as an unsupported query parameter. Retrieve the broader event set and filter client-side instead. `dataValues` is returned as a list-column — use `tidyr::unnest_wider()/tidyr::unnest_longer()` to flatten it further.

Value

A tibble of events, or NULL if none were found.

See Also

`get_tracked_entities()`, `get_enrollments()`

Other tracker functions: `get_enrollments()`, `get_relationships()`, `get_tracked_entities()`, `tracked_entity_filter()`

Examples

```
# All events for a program stage at a given org unit
get_events(program_stage = 'mj1stImcUCi',
           org_unit = 'NRcrkSgDX5G',
           org_unit_mode = 'DESCENDANTS')
```

get_event_analytics	<i>Retrieves Aggregated Event Analytics Data from DHIS2</i>
---------------------	---

Description

[Experimental] `get_event_analytics()` retrieves aggregated, dimensional analytics over Tracker events from DHIS2's `analytics/events/query/{program}` endpoint — as opposed to `get_events()`, which returns raw individual event records from the Tracker API. Use this when you want counts/aggregates across dimensions (e.g. events by org unit and period), not the underlying records themselves.

Usage

```
get_event_analytics(
  program,
  ...,
  return_type = c("uid", "name"),
  page_size = 1000,
  retry = 2,
  verbosity = 0,
  timeout = 60,
  auth = NULL,
  call = caller_env()
)
```

Arguments

<code>program</code>	A program id. Used as a URL path segment, not a query parameter.
<code>...</code>	One or more <code>analytics_dimension()</code> parameters (e.g. <code>ou/pe</code> dimensions via <code>%d/%f</code>), and/or other query parameters supported by your DHIS2 instance's Tracker analytics query API.
<code>return_type</code>	Optional. 'uid' (default) or 'name' for the identifier scheme used in the response.
<code>page_size</code>	Number of records to request per page (default 1000).
<code>retry</code>	Number of times to retry the API call in case of failure (defaults to 2).
<code>verbosity</code>	Level of HTTP information to print during the call.
<code>timeout</code>	Maximum number of seconds to wait for the API response.
<code>auth</code>	Optional. The authentication object.
<code>call</code>	The caller environment.

Details

The response shares the same headers/rows shape as `get_analytics()`, confirmed live against a public DHIS2 demo instance. Pagination is automatic, but confirmed live to work differently from every other paginated function in this package: without an explicit page size, this endpoint silently returns only the first 50 rows, with no error or warning — a real risk if you call the underlying DHIS2 endpoint directly instead of through this function.

Value

A tibble of aggregated event analytics data, or NULL if none was retrieved.

See Also

[get_events\(\)](#) for raw individual event records, [get_enrollment_analytics\(\)](#) for the enrollment equivalent, [get_analytics\(\)](#) for aggregate (non-Tracker) analytics.

Examples

```
# Events for a program, by org unit, over the last 12 months
get_event_analytics(program = 'PREnRHSp3be',
                    ou %.d% 'USER_ORGUNIT',
                    pe %.d% 'LAST_12_MONTHS')
```

```
get_event_analytics_aggregate
```

Get Aggregated (Pivot-Style) Event Analytics from DHIS2

Description

[Experimental] `get_event_analytics_aggregate()` retrieves totals over Tracker events from DHIS2's `analytics/events/aggregate/{program}` endpoint. This is a genuinely different endpoint from [get_event_analytics\(\)](#)'s `analytics/events/query/{program}`: query returns line-list style rows (one row per matched dimension combination, event-count driven), while aggregate returns pivot-table style totals, and supports a stage parameter to scope to one program stage.

Usage

```
get_event_analytics_aggregate(
  program,
  stage = NULL,
  ...,
  return_type = c("uid", "name"),
  retry = 2,
  verbosity = 0,
  timeout = 60,
  auth = NULL,
  call = caller_env()
)
```

get_file_resources	<i>Get File Resources Metadata from a DHIS2 Instance</i>
--------------------	--

Description

[Experimental] `get_file_resources()` lists the metadata of files stored in a DHIS2 instance — icons, and files/images attached to tracker attribute values or data values — from the `fileResources` endpoint. This retrieves metadata only (name, content type, size); it does not download file contents.

Usage

```
get_file_resources(
  ...,
  retry = 2,
  verbosity = 0,
  timeout = 60,
  auth = NULL,
  call = caller_env()
)
```

Arguments

...	<code>metadata_filter()</code> parameters, or their infix-operator shorthand, to filter the results, and/or other query parameters supported by your DHIS2 instance's <code>fileResources</code> endpoint.
<code>retry</code>	Number of times to retry the API call in case of failure (defaults to 2).
<code>verbosity</code>	Level of HTTP information to print during the call.
<code>timeout</code>	Maximum number of seconds to wait for the API response.
<code>auth</code>	Optional. The authentication object.
<code>call</code>	The caller environment.

Value

A tibble of file resources (`id`, `name`, `contentType`, `contentLength`), or `NULL` if none were found. Confirmed live against a public DHIS2 demo instance.

Examples

```
get_file_resources(contentType %like% 'image')
```

get_geo_features	<i>Get Organisation Unit Geographic Features</i>
------------------	--

Description

[Experimental] get_geo_features() retrieves the geographic features (coordinates/ boundaries) of organisation units from the geoFeatures endpoint — useful for mapping. Confirmed live against a public DHIS2 demo instance.

Usage

```
get_geo_features(
  org_units,
  ...,
  retry = 2,
  verbosity = 0,
  timeout = 60,
  auth = NULL,
  call = caller_env()
)
```

Arguments

org_units	A vector of organisation unit ids, or a DHIS2 org-unit keyword such as 'LEVEL-2' or 'USER_ORGUNIT'.
...	Other query parameters supported by your DHIS2 instance's geoFeatures endpoint.
retry	Number of times to retry the API call in case of failure (defaults to 2).
verbosity	Level of HTTP information to print during the call.
timeout	Maximum number of seconds to wait for the API response.
auth	Optional. The authentication object.
call	The caller environment.

Details

Confirmed live that org_units must be sent as a ou:-prefixed dimension value (e.g. ou:LEVEL-2 or ou:<uid1>;<uid2>) — a bare comma/semicolon-separated list of ids, without the ou: prefix, is rejected. A dimensions field in the raw response (category-dimension filters applied to the query; empty in every case tested) is dropped before returning, since an empty list-column in every row otherwise collapses the whole result to 0 rows.

Value

A tibble with one row per organisation unit: id, name, code, level, type (point or polygon), parent_id, parent_name, parent_graph, coordinates (a GeoJSON-style coordinate string), has_coordinates_down, and has_coordinates_up. NULL if none were found.

Examples

```
get_geo_features(org_units = 'LEVEL-2')
```

get_metadata	<i>Get Metadata from a DHIS2 Instance</i>
--------------	---

Description

get_metadata retrieves metadata for a specified endpoint of a DHIS2 instance.

Usage

```
get_metadata(
  endpoint,
  ...,
  fields = c("id", "name"),
  retry = 2,
  verbosity = 0,
  timeout = 60,
  call = caller_env()
)
```

Arguments

endpoint	The DHIS2 API endpoint for the metadata of interest (e.g., dataElements, organisationUnits endpoints).
...	One or more metadata_filter() parameters in key-value pairs.
fields	The specific columns to be returned in the data frame.
retry	Number of times to retry the API call in case of failure (defaults to 2).
verbosity	Level of HTTP information to print during the call: • 0: No output • 1: Show headers • 2: Show headers and bodies • 3: Show headers, bodies, and CURL status message.
timeout	Maximum number of seconds to wait for the DHIS2 API response.
call	The caller environment

Value

A tibble containing the DHIS2 metadata response.

Examples

```
# Get the categories metadata
get_metadata('categories')

# Get the datasets metadata with fields 'id,name,organisationUnits' and filter
# only the datasets with id 'VEM58nY22s0'
get_metadata('dataSets',
             fields = 'id,name,organisationUnits[id,name,path]',
             id %.eq% 'VEM58nY22s0')

# Get data elements filtered by dataElementGroups id
get_metadata('dataElements',
             dataElementGroups.id %.eq% 'WdtYaV7o8QV',
             fields = ':all')
```

```
get_organisations_by_level
```

Get Organisations by Level

Description

[Experimental] `get_organisations_by_level()` is an experimental function that retrieves the organisation units along with their parent units.

Usage

```
get_organisations_by_level(
  level = 1,
  org_ids = NULL,
  auth = NULL,
  call = caller_env()
)
```

Arguments

<code>level</code>	An integer specifying the desired organisation level (default level 1).
<code>org_ids</code>	Optional. A vector of organisation identifiers whose details are being retrieved.
<code>auth</code>	Optional. The authentication object
<code>call</code>	The call environment.

Value

A tibble containing the organisation units and their parent units up to the specified level. For each ancestor level, both a name column (e.g. county) and an id column (e.g. county_id) are included, so results can be joined back to other org-unit-keyed data by id rather than name — two different org units at the same level can share a name, a real, known DHIS2 data-quality issue.

Examples

```
# Fetch all the organisation units metadata
organisations <- get_organisations_by_level(level = 2)
organisations
```

get_relationships	<i>Get Relationships from a DHIS2 Instance</i>
-------------------	--

Description

[Experimental] `get_relationships()` retrieves Tracker relationships — links between two tracker objects (tracked entities, enrollments, or events), such as an index case and a household contact — from DHIS2's Tracker API.

Usage

```
get_relationships(
  tracked_entity = NULL,
  enrollment = NULL,
  event = NULL,
  ...,
  fields = c("relationship", "relationshipType", "from", "to", "createdAt"),
  page_size = 500,
  retry = 2,
  verbosity = 0,
  timeout = 60,
  auth = NULL,
  call = caller_env()
)
```

Arguments

<code>tracked_entity</code> , <code>enrollment</code> , <code>event</code>	Exactly one of these three must be provided: a tracked entity, enrollment, or event id to retrieve relationships for.
<code>...</code>	Other query parameters supported by your DHIS2 instance's Tracker API.
<code>fields</code>	The DHIS2 field-selector for the columns to return.
<code>page_size</code>	Number of records to request per page (default 500).
<code>retry</code>	Number of times to retry the API call in case of failure (defaults to 2).
<code>verbosity</code>	Level of HTTP information to print during the call.
<code>timeout</code>	Maximum number of seconds to wait for the DHIS2 API response.
<code>auth</code>	Optional. The authentication object.
<code>call</code>	The caller environment.

Details

DHIS2 requires a relationships query to be scoped by exactly one of `tracked_entity`, `enrollment`, or `event` — confirmed live against a public DHIS2 demo instance, where omitting all three returns the exact error "Missing required parameter 'trackedEntity', 'enrollment' or 'event'.". Pagination (nested pager, same as [get_tracked_entities\(\)](#)/[get_events\(\)](#)/[get_enrollments\(\)](#)) was also confirmed live, but the demo instance tested had no relationship data configured on any program, so the shape of a populated relationship's `from/to` fields is not independently verified here — inspect the result and adjust fields as needed for your instance.

Value

A tibble of relationships, or NULL if none were found.

See Also

[get_tracked_entities\(\)](#), [get_events\(\)](#), [get_enrollments\(\)](#)

Other tracker functions: [get_enrollments\(\)](#), [get_events\(\)](#), [get_tracked_entities\(\)](#), [tracked_entity_filter\(\)](#)

Examples

```
# Relationships for a specific tracked entity
get_relationships(tracked_entity = 'qkU5JI6SQcd')
```

`get_sql_views`*Get SQL Views Metadata from a DHIS2 Instance*

Description

[Experimental] `get_sql_views()` lists the SQL views configured on a DHIS2 instance — pre-defined, admin-authored SQL queries exposed as a data endpoint. Use [get_sql_view_data\(\)](#) to retrieve the actual data for one of them.

Usage

```
get_sql_views(
  ...,
  retry = 2,
  verbosity = 0,
  timeout = 60,
  auth = NULL,
  call = caller_env()
)
```

Arguments

...	metadata_filter() parameters, or their infix-operator shorthand, to filter the results, and/or other query parameters supported by your DHIS2 instance's sqlViews endpoint.
retry	Number of times to retry the API call in case of failure (defaults to 2).
verbosity	Level of HTTP information to print during the call.
timeout	Maximum number of seconds to wait for the API response.
auth	Optional. The authentication object.
call	The caller environment.

Value

A tibble of SQL views (id, name, type — 'VIEW' for a materialized view queryable directly, or 'QUERY' for one that must be executed first), or NULL if none were found. Confirmed live against a public DHIS2 demo instance.

See Also

[get_sql_view_data\(\)](#) for retrieving a view's data.

Examples

```
get_sql_views()
```

get_sql_view_data	<i>Get Data from a DHIS2 SQL View</i>
-------------------	---------------------------------------

Description

[Experimental] [get_sql_view_data\(\)](#) retrieves the data produced by a SQL view identified by [get_sql_views\(\)](#) — a predefined, admin-authored SQL query exposed as a data endpoint.

Usage

```
get_sql_view_data(
  sql_view,
  variables = NULL,
  ...,
  retry = 2,
  verbosity = 0,
  timeout = 60,
  auth = NULL,
  call = caller_env()
)
```

Arguments

sql_view	A SQL view id.
variables	Optional. A named character vector of variable substitutions for a parameterised SQL view, e.g. <code>c(orgUnit = 'ImspTQPwCqd')</code> .
...	Other query parameters supported by your DHIS2 instance's <code>sqlViews</code> data endpoint.
retry	Number of times to retry the API call in case of failure (defaults to 2).
verbosity	Level of HTTP information to print during the call.
timeout	Maximum number of seconds to wait for the API response.
auth	Optional. The authentication object.
call	The caller environment.

Details

A view of type 'QUERY' (see [get_sql_views\(\)](#)) is executed on demand; a 'VIEW' is queried directly. **This function's response shape has not been independently verified live:** every SQL view on the public demo instance used to verify `khizr`'s other functions returned 403 Forbidden ("not authorised") or 409 Conflict (referencing an analytics table this particular demo hadn't built) for this account — both account/instance issues rather than request errors. The shape below follows DHIS2's documented `listGrid` structure (shared with a few other legacy DHIS2 endpoints), reusing the same headers/rows parsing as [get_analytics\(\)](#); confirm it against your own instance before relying on this function in production.

Value

A tibble of the SQL view's result rows, or NULL if none were retrieved.

See Also

[get_sql_views\(\)](#) for listing available SQL views.

Examples

```
## Not run:
views <- get_sql_views()
get_sql_view_data(views$id[1])

## End(Not run)
```

get_system_info	<i>Get DHIS2 Instance System Information</i>
-----------------	--

Description

[Experimental] get_system_info() retrieves information about the connected DHIS2 instance itself — version, revision, build time, server date/time zone, and similar — from the system/info endpoint. Useful for checking compatibility before relying on version-specific behaviour.

Usage

```
get_system_info(auth = NULL, call = caller_env())
```

Arguments

auth	Optional. The authentication object.
call	The caller environment.

Value

A named list of system information fields (e.g. version, revision, buildTime, serverDate, contextPath), as returned by DHIS2. Confirmed live against a public DHIS2 demo instance.

See Also

[khis_api_version\(\)](#) for the API version pinned on the current credentials.

Examples

```
get_system_info()
```

get_tracked_entities	<i>Get Tracked Entities from a DHIS2 Instance</i>
----------------------	---

Description

[Experimental] get_tracked_entities() retrieves tracked entity instances (e.g. patients, clients) from the DHIS2 Tracker API.

Usage

```

get_tracked_entities(
  program = NULL,
  ...,
  tracked_entity_type = NULL,
  tracked_entities = NULL,
  org_units = NULL,
  org_unit_mode = NULL,
  updated_after = NULL,
  updated_before = NULL,
  enrolled_after = NULL,
  enrolled_before = NULL,
  occurred_after = NULL,
  occurred_before = NULL,
  fields = c("trackedEntity", "trackedEntityType", "orgUnit", "createdAt", "updatedAt",
    "inactive"),
  page_size = 500,
  retry = 2,
  verbosity = 0,
  timeout = 60,
  auth = NULL,
  call = caller_env()
)

```

Arguments

<code>program</code>	Optional. A program id to scope the query to.
<code>...</code>	One or more tracked_entity_filter() attribute filters (or its infix operators, e.g. <code>w75KJ2mc4zz % .teq% ' John '</code>), and/or other query parameters supported by your DHIS2 instance's Tracker API. Build filters with tracked_entity_filter() and its own infix operators, not metadata_filter() or its infix operators (<code>% .eq%</code> , <code>% .in%</code> , etc.) — those target the DHIS2 metadata API's larger operator set and, for <code>in/!in</code> , a different value-joining convention; using them here raises an error rather than silently sending malformed filter syntax.
<code>tracked_entity_type</code>	Optional. A tracked entity type id to scope the query to.
<code>tracked_entities</code>	Optional. A vector of specific tracked entity ids to retrieve.
<code>org_units</code>	Optional. A vector of organisation unit ids to scope the query to.
<code>org_unit_mode</code>	Optional. One of "SELECTED", "CHILDREN", "DESCENDANTS", "ACCESSIBLE", "CAPTURE", "ALL", controlling how <code>org_units</code> is interpreted. DHIS2 defaults to "ACCESSIBLE" when <code>org_units</code> is not provided, and "SELECTED" when it is.
<code>updated_after</code> , <code>updated_before</code>	Optional. ISO-8601 date or datetime strings bounding the tracked entity's last-updated timestamp.

enrolled_after, enrolled_before	Optional. ISO-8601 date or datetime strings bounding the tracked entity's enrollment date.
occurred_after, occurred_before	Optional. ISO-8601 date or datetime strings bounding the tracked entity's enrollment incident/occurred date.
fields	The DHIS2 field-selector for the columns to return.
page_size	Number of records to request per page (default 500).
retry	Number of times to retry the API call in case of failure (defaults to 2).
verbosity	Level of HTTP information to print during the call.
timeout	Maximum number of seconds to wait for the DHIS2 API response.
auth	Optional. The authentication object.
call	The caller environment.

Details

DHIS2 requires a tracked entities query to be scoped by at least one of program, tracked_entity_type, tracked_entities, or org_units; an unscoped query will be rejected by the server. Nested repeating data (e.g. attributes, enrollments) is returned as a list-column — use `tidyr::unnest_wider()/tidyr::unnest_tidy()` to flatten it further.

Value

A tibble of tracked entities, or NULL if none were found.

See Also

[tracked_entity_filter\(\)](#) for filtering on attribute values.

Other tracker functions: [get_enrollments\(\)](#), [get_events\(\)](#), [get_relationships\(\)](#), [tracked_entity_filter\(\)](#)

Examples

```
# All tracked entities enrolled in a program at a given org unit
get_tracked_entities(program = 'PREnRHSp3be',
                     org_units = 'IWp9dQGM0bS',
                     org_unit_mode = 'DESCENDANTS')

# Tracked entities whose attribute mTYYajEh1PY contains "John"
get_tracked_entities(program = 'PREnRHSp3be',
                     org_units = 'IWp9dQGM0bS',
                     tracked_entity_filter('mTYYajEh1PY', 'like', 'John'))
```

`get_validation_results`*Get Validation Rule Results from DHIS2*

Description

[Experimental] `get_validation_results()` retrieves violated validation rules for an organisation unit and period range from DHIS2's `validationResults` endpoint — useful for data-quality review.

Usage

```
get_validation_results(  
  org_units,  
  start_date,  
  end_date,  
  ...,  
  retry = 2,  
  verbosity = 0,  
  timeout = 60,  
  auth = NULL,  
  call = caller_env()  
)
```

Arguments

<code>org_units</code>	A vector of organisation unit ids to scope the query to.
<code>start_date, end_date</code>	ISO-8601 dates bounding the query.
<code>...</code>	Other query parameters supported by your DHIS2 instance's <code>validationResults</code> endpoint (e.g. <code>vrg</code> to scope to a validation rule group).
<code>retry</code>	Number of times to retry the API call in case of failure (defaults to 2).
<code>verbosity</code>	Level of HTTP information to print during the call.
<code>timeout</code>	Maximum number of seconds to wait for the API response.
<code>auth</code>	Optional. The authentication object.
<code>call</code>	The caller environment.

Details

Confirmed live against a public DHIS2 demo instance that this endpoint (`/api/validationResults`, not `/api/analytics/validationResults`, which 404s) accepts these parameters and returns a valid, empty result — the instance tested had no stored validation violations for the org units and periods tried, so the shape of a populated result is not independently verified.

Value

A tibble of validation results (organisation unit, period, validation rule, and the left/right side values that violated it), or NULL if none were found.

Examples

```
get_validation_results(org_units = 'IWp9dQGM0bS',
  start_date = '2023-01-01',
  end_date = '2023-12-31')
```

khisr-configuration	<i>khisr Configuration</i>
---------------------	----------------------------

Description

Some aspects of khisr behaviour can be controlled via an option.

Temporarily suppress messages by enabling quiet mode within the provided code block.

Temporarily suppress messages within the specified environment.

Usage

```
with_khis_quiet(code)
```

```
local_khis_quiet(env = parent.frame())
```

Arguments

code	Code to execute quietly
env	The environment to use for scoping.

Value

No return value, called for side effects

No return value, called for side effects

No return value, called for side effects.

Messages

The `khis_quiet` option can be used to suppress messages from khisr. By default, khisr always messages, i.e. it is *not* quiet.

set `khis_quiet` to TRUE to suppress message, by one of these means, in order of decreasing scope:

- Put `options(khis_quiet = TRUE)` in the start-up file, such as `.Rprofile`, or in your R script.
- Use `local_khis_quiet()` to silence khisr in a specific scope.

- Use `with_khis_quiet` to run small bit of code silently.

`local_khis_quiet` and `with_khis_quiet` follow the conventions of the `withr` package.

Examples

```
## Not run:
# message: "The credentials have been set."
khis_cred(username = 'username',
           password = 'password',
           server = 'https://<dhis2-instance>')

# suppress messages for a small amount of code
with_khis_quiet(
  khis_cred(username = 'username',
            password = 'password',
            server = 'https://<dhis2-instance>')
)

## End(Not run)

## Not run:
# message: "The credentials have been set."
khis_cred(username = 'username',
           password = 'password',
           server = 'https://<dhis2-instance>')

# suppress messages in a specific scope
local_khis_quiet()

# no message
khis_cred(username = 'username',
           password = 'password',
           server = 'https://<dhis2-instance>')

# clear credentials
khis_cred_clear()

## End(Not run)
```

khis_api_version

Retrieve the Configured DHIS2 API Version

Description

This function returns the API version requests are pinned to, as set via the `api_version` argument of `khis_cred()`. Returns NULL when no version is pinned, meaning requests use the server's own default version.

Usage

```
khis_api_version(auth = NULL)
```

Arguments

auth (Optional) An auth object containing the DHIS2 credentials. If not provided, the function retrieves the API version from the global auth object.

Value

The pinned DHIS2 API version as a string, or NULL if not set.

See Also

Other credential functions: [khis_base_url\(\)](#), [khis_cred\(\)](#), [khis_cred_clear\(\)](#), [khis_display_name\(\)](#), [khis_has_cred\(\)](#), [khis_username\(\)](#)

Examples

```
## Not run:
  khis_cred(username = 'DHIS2 username',
            password = 'DHIS2 password',
            server = 'https://<dhis2-instance>',
            api_version = '40')

  # Retrieve the pinned API version (expect '40')
  khis_api_version()

## End(Not run)
```

khis_base_url

Retrieve the Configured DHIS2 API Base URL

Description

This function returns the base URL for the DHIS2 API from the provided auth object, or falls back to the global auth credentials if auth is not provided.

Usage

```
khis_base_url(auth = NULL)
```

Arguments

auth (Optional) An auth object containing the DHIS2 credentials. If not provided, the function retrieves the base URL from the global auth object.

Value

The DHIS2 base URL as a string, or NULL if no credentials are available.

See Also

Other credential functions: [khis_api_version\(\)](#), [khis_cred\(\)](#), [khis_cred_clear\(\)](#), [khis_display_name\(\)](#), [khis_has_cred\(\)](#), [khis_username\(\)](#)

Examples

```
## Not run:
# Set the credentials using the global .auth object
khis_cred(username = 'DHIS2 username',
           password = 'DHIS2 password',
           server = 'https://<dhis2-instance>')

# Retrieve the DHIS2 instance API base URL (expect 'https://<dhis2-instance>')
khis_base_url()

# Clear credentials
khis_cred_clear()

# Retrieve the base URL again (expect 'NULL')
khis_base_url()

## End(Not run)
```

khis_cred	<i>Sets DHIS2 Credentials</i>
-----------	-------------------------------

Description

`khis_cred()` sets the credentials for accessing a DHIS2 instance.

Usage

```
khis_cred(
  username = NULL,
  password = NULL,
  token = NULL,
  server = NULL,
  api_version = NULL,
  config_path = NULL,
  base_url = deprecated()
)
```

Arguments

username	The DHIS2 username. Only required if neither config_path nor token is provided.
password	The DHIS2 password. Only required if neither config_path nor token is provided.
token	A DHIS2 Personal Access Token , as an alternative to username/password. DHIS2 recommends tokens over Basic Authentication for scripts and integrations. Cannot be combined with username/password or config_path.
server	The server URL of the DHIS2 instance. Only required if configuration file not provided.
api_version	Optional. Pins requests to a specific DHIS2 API version (e.g. "40"), so calls hit <server>/api/40/<endpoint> instead of <server>/api/<endpoint>. Useful for guarding against behavioural differences between DHIS2 core versions across instances. Defaults to the server's own default API version when not set.
config_path	An optional path to a configuration file containing either username/password or a token. This is considered more secure than providing credentials directly in code.
base_url	Deprecated. The base URL of the DHIS2 instance. Use server instead.

Details

This function allows you to set the credentials for interacting with a DHIS2 server. You can provide username/password directly, a token directly, or specify a path to a configuration file containing either. Using a configuration file is recommended for improved security as it prevents credentials from being stored directly in your code.

Token authentication sends Authorization: ApiToken <token>, confirmed by generating a real token (POST /api/apiToken) against a live public DHIS2 demo instance and using it to authenticate and retrieve real data; like Basic Authentication's username/password, the token is redacted and never printed in verbose/debug request output.

Value

Auth object

See Also

Other credential functions: [khis_api_version\(\)](#), [khis_base_url\(\)](#), [khis_cred_clear\(\)](#), [khis_display_name\(\)](#), [khis_has_cred\(\)](#), [khis_username\(\)](#)

Examples

```
## Not run:
# Option 1: Using a configuration file (recommended)
# Assuming a configuration file named "credentials.json":
khis_cred(config_path = "path/to/credentials.json")

# Option 2: Providing username/password directly (less secure)
```

```

khis_cred(username = "your_username",
           password = "your_password",
           server='https://<dhis2-instance>')

# Option 3: Providing a Personal Access Token directly (less secure)
khis_cred(token = "d2pat_...",
           server = 'https://<dhis2-instance>')

## End(Not run)

```

khis_cred_clear	<i>Clear the Credentials from Memory</i>
-----------------	--

Description

This function clears the DHIS2 credentials from memory. If an auth object is provided, it clears the credentials from that object. If no auth object is provided, it clears the global auth credentials.

Usage

```
khis_cred_clear(auth = NULL)
```

Arguments

auth	(Optional) An authentication object from which to clear credentials. If not provided, the credentials in the global auth object will be cleared.
------	--

Value

No return value, called for side effects.

See Also

Other credential functions: [khis_api_version\(\)](#), [khis_base_url\(\)](#), [khis_cred\(\)](#), [khis_display_name\(\)](#), [khis_has_cred\(\)](#), [khis_username\(\)](#)

Examples

```

# Clear credentials from the global .auth object
khis_cred_clear()

```

khis_display_name	<i>Retrieve the Configured Display Name</i>
-------------------	---

Description

This function returns the display name from the configured profile in the provided auth object. If auth is not provided, it falls back to the global auth credentials.

Usage

```
khis_display_name(auth = NULL)
```

Arguments

auth	(Optional) An auth object containing DHIS2 credentials. If not provided, the function retrieves the display name from the global auth object.
------	---

Value

The display name as a string, or NULL if no profile or display name is available.

See Also

Other credential functions: [khis_api_version\(\)](#), [khis_base_url\(\)](#), [khis_cred\(\)](#), [khis_cred_clear\(\)](#), [khis_has_cred\(\)](#), [khis_username\(\)](#)

Examples

```
## Not run:
# Set the credentials using global .auth object
khis_cred(username = 'DHIS2 username',
           password = 'DHIS2 password',
           server = 'https://<dhis2-instance>')

# Retrieve the display name from the global .auth profile
khis_display_name()

# Clear credentials
khis_cred_clear()

# Retrieve the display name again (expect 'NULL')
khis_display_name()

## End(Not run)
```

khis_has_cred*Check if DHIS2 Credentials are Available*

Description

This function checks whether valid credentials are available either in the provided auth object or in the global auth credentials object.

Usage

```
khis_has_cred(auth = NULL)
```

Arguments

auth (Optional) An auth object containing DHIS2 credentials. If not provided, the function will check the global auth object for credentials.

Value

A boolean value indicating whether valid credentials are available.

See Also

Other credential functions: [khis_api_version\(\)](#), [khis_base_url\(\)](#), [khis_cred\(\)](#), [khis_cred_clear\(\)](#), [khis_display_name\(\)](#), [khis_username\(\)](#)

Examples

```
## Not run:
# Set the credentials using global .auth object
khis_cred(username = 'DHIS2 username',
           password = 'DHIS2 password',
           server = 'https://<dhis2-instance>')

# Check if credentials are available. Should return TRUE
khis_has_cred()

# Clear global credentials
khis_cred_clear()

# Check if credentials are available. Should return FALSE
khis_has_cred()

## End(Not run)
```

khis_username	<i>Retrieve the Configured Username</i>
---------------	---

Description

This function returns the username from the configured credentials. If an auth object is provided, it retrieves the username from that object. Otherwise, it retrieves the username from the global auth object.

Usage

```
khis_username(auth = NULL)
```

Arguments

auth	(Optional) An auth object. If not provided, the function will retrieve the username from the global auth credentials.
------	---

Value

The username as a string, or NULL if no credentials are available, or if the credentials were set with token rather than username/password (see [khis_cred\(\)](#)).

See Also

Other credential functions: [khis_api_version\(\)](#), [khis_base_url\(\)](#), [khis_cred\(\)](#), [khis_cred_clear\(\)](#), [khis_display_name\(\)](#), [khis_has_cred\(\)](#)

Examples

```
## Not run:
# Set the credentials using global .auth object
khis_cred(username = 'DHIS2 username',
           password = 'DHIS2 password',
           server = 'https://<dhis2-instance>')

# View the username (expect 'DHIS2 username')
khis_username()

# Clear credentials
khis_cred_clear()

# View the username (expect 'NULL')
khis_username()

## End(Not run)
```

metadata-filter	<i>Metadata Filter</i>
-----------------	------------------------

Description

Formats a metadata filter to DHIS 2 comparison operators.

Usage

```
metadata_filter(property, operator, values, call = caller_env())
```

```
property %.eq% values
```

```
property %.ieq% values
```

```
property %.~eq% values
```

```
property %.ne% values
```

```
property %.Like% values
```

```
property %.~Like% values
```

```
property %.^Like% values
```

```
property %.~^Like% values
```

```
property %.Like$% values
```

```
property %.~Like$% values
```

```
property %.like% values
```

```
property %.~like% values
```

```
property %.^like% values
```

```
property %.~^like% values
```

```
property %.like$% values
```

```
property %.~like$% values
```

```
property %.gt% values
```

```
property %.ge% values
```

property %.lt% values

property %.le% values

property %.token% values

property %~token% values

property %.in% values

property %~in% values

Arguments

property	The property on the metadata you want to filter on
operator	The comparison operator you want to perform
values	The value to check against
call	The caller environment.

Details

To filter the metadata there are several filter operations that can be applied to the returned list of metadata. The format of the filter itself is straight-forward and follows the pattern `property:operator:value`, where `property` is the property on the metadata you want to filter on, `operator` is the comparison operator you want to perform and `value` is the value to check against (not all operators require value).

To view the operator see [DHIS 2 Operator](#)

- `%.eq%` - Equality
- `%.ieq%` - Case insensitive string, match exact
- `%.~eq%` - Inequality
- `%.ne%` - Inequality
- `%.Like%` - Case sensitive string, match anywhere
- `%.~Like%` - Case sensitive string, not match anywhere
- `%.^Like%` - Case sensitive string, match start
- `%.~^Like%` - Case sensitive string, not match start
- `%.Like$%` - Case sensitive string, match end
- `%.~Like$%` - Case sensitive string, not match end
- `%.like%` - Case insensitive string, match anywhere
- `%.~like%` - Case insensitive string, not match anywhere
- `%.^like%` - Case insensitive string, match start
- `%.~^like%` - Case insensitive string, not match start
- `%.like$%` - Case insensitive string, match end
- `%.~like$%` - Case insensitive string, not match end
- `%.gt%` - Greater than
- `%.ge%` - Greater than or equal
- `%.lt%` - Less than

- `%.le%` - Less than or equal
- `%.token%` - Match on multiple tokens in search property
- `%.~token%` - Not match on multiple tokens in search property
- `%.in%` - Find objects matching 1 or more values
- `%.~in%` - Find objects not matching 1 or more values

Value

A spliced list with filter in the format `property:operator:value`

Examples

```
# Generate an equality filter
id %.eq% 'element_id'

# Finding multiple ids
'id' %.in% c('id1', 'id2', 'id3')

# Get all data elements which have a data set with id ID1
'dataSetElements.dataSet.id' %.eq% 'ID1'

# get data elements which are members of the ANC data element group
'dataElementGroups.id' %.eq% 'qfxEYY9xA16'

# Get data elements which have any option set
metadata_filter('optionSet', '!=null', NULL)
```

metadata-helpers

DHIS2 Metadata Helper Functions

Description

These functions simplify retrieving data from specific DHIS2 API endpoints using [get_metadata\(\)](#).

Usage

```
get_categories(...)

get_category_combos(...)

get_category_option_combos(...)

get_category_option_group_sets(...)

get_category_option_groups(...)

get_category_options(...)
```

```
get_data_element_group_sets(...)
get_data_element_groups(...)
get_data_elements(...)
get_data_sets(...)
get_user_groups(...)
get_indicator_group_sets(...)
get_indicator_groups(...)
get_indicators(...)
get_option_group_sets(...)
get_option_groups(...)
get_option_sets(...)
get_options(...)
get_organisation_unit_groupsets(...)
get_organisation_unit_groups(...)
get_organisation_units(...)
get_organisation_unit_levels(...)
get_dimensions(...)
get_period_types(...)
get_programs(...)
get_program_stages(...)
get_tracked_entity_types(...)
get_tracked_entity_attributes(...)
get_relationship_types(...)
get_user_profile()
```

Arguments

... Arguments passed on to [get_metadata](#)

fields The specific columns to be returned in the data frame.

retry Number of times to retry the API call in case of failure (defaults to 2).

verbosity Level of HTTP information to print during the call:

- 0: No output
- 1: Show headers
- 2: Show headers and bodies
- 3: Show headers, bodies, and CURL status message.

timeout Maximum number of seconds to wait for the DHIS2 API response.

call The caller environment

Value

A tibble containing the DHIS2 metadata response.

See Also

[get_tracked_entities\(\)](#), [get_events\(\)](#), [get_enrollments\(\)](#) for retrieving tracker data (as opposed to the metadata describing programs, program stages, and tracked entity types/attributes retrieved here).

Examples

```
# Get all organisation units
get_organisation_units()

# Get all data elements
get_data_elements()

# Get data elements by element ids
get_data_elements(id %.in% c('lYsfXxCw6Qi', 'GxlrIgMyEf4'))

# Get datasets by name with the word 'malaria'
get_data_sets(name %.like% 'malaria')

# Get all tracker programs
get_programs()
```

tracked_entity_filter *Tracked Entity Attribute Filter*

Description

Formats a filter on a tracked entity attribute for the DHIS2 Tracker API's `trackedEntities` endpoint, in the `attribute:operator:value` form it expects. This is a separate, smaller operator set from `metadata_filter()` — the two are not interchangeable, and the `in` operator uses a different value-joining convention (semicolons, no brackets).

Usage

```
tracked_entity_filter(attribute, operator, values, call = caller_env())
```

```
attribute %.teq% values
```

```
attribute %.tne% values
```

```
attribute %.tgt% values
```

```
attribute %.tge% values
```

```
attribute %.tlt% values
```

```
attribute %.tle% values
```

```
attribute %.tlike% values
```

```
attribute %.tsw% values
```

```
attribute %.tew% values
```

```
attribute %.tin% values
```

Arguments

attribute	The tracked entity attribute id to filter on.
operator	The comparison operator to apply.
values	The value(s) to compare against. Not required for <code>null</code> and <code>!null</code> .
call	The caller environment.

Details

As documented for the DHIS2 Tracker API, the supported operators are:

- `eq` - `%.teq%` - Equality

- `ge` - `%.tge%` - Greater than or equal
- `gt` - `%.tgt%` - Greater than
- `le` - `%.tle%` - Less than or equal
- `lt` - `%.tlt%` - Less than
- `ne` - `%.tne%` - Inequality
- `like` - `%.tlike%` - Match anywhere
- `sw` - `%.tsw%` - Starts with
- `ew` - `%.tew%` - Ends with
- `in` - `%.tin%` - Match one or more values
- `null` - (no infix form; call `tracked_entity_filter(attr, 'null', NULL)`) - Attribute has no value
- `!null` - (no infix form; call `tracked_entity_filter(attr, '!null', NULL)`) - Attribute has a value

The infix operators are shorthand for the equivalent `tracked_entity_filter()` call, e.g. `mTYYajEhlPY %.teq% 'John'` is equivalent to `tracked_entity_filter('mTYYajEhlPY', 'eq', 'John').null!/null` have no infix form since an infix operator needs a right-hand value.

This filter is only documented for the `trackedEntities` endpoint. DHIS2's published Tracker API docs do not describe an equivalent way to filter events or enrollments, so `get_events()` and `get_enrollments()` reject a filter argument outright rather than silently sending it as unsupported query syntax.

`metadata_filter()` and its infix operators (`%.eq%`, `%.in%`, etc.) are for a different DHIS2 API and are not interchangeable with this function — passing one to `get_tracked_entities()` raises an error, since several metadata operators (e.g. `ieq`, `token`, the anchored `like` variants) and the `in`/`!in` value-joining convention (comma-bracketed vs semicolon) don't match what the Tracker API expects.

Value

A spliced list with `filter` in the format `attribute:operator:value`, suitable for passing straight into `get_tracked_entities()`.

See Also

Other tracker functions: `get_enrollments()`, `get_events()`, `get_relationships()`, `get_tracked_entities()`

Examples

```
# Tracked entities where attribute mTYYajEhlPY equals "John"
tracked_entity_filter('mTYYajEhlPY', 'eq', 'John')

# Tracked entities where attribute mTYYajEhlPY is one of several values
tracked_entity_filter('mTYYajEhlPY', 'in', c('John', 'Jane'))

# Equivalent, using the infix operator
mTYYajEhlPY %.teq% 'John'
```

Index

* credential functions

- khis_api_version, 41
- khis_base_url, 42
- khis_cred, 43
- khis_cred_clear, 45
- khis_display_name, 46
- khis_has_cred, 47
- khis_username, 48

* tracker functions

- get_enrollments, 18
- get_events, 23
- get_relationships, 32
- get_tracked_entities, 36
- tracked_entity_filter, 54
- %.Like%(metadata-filter), 49
- %.Like%(metadata-filter), 49
- %.^Like%(metadata-filter), 49
- %.^like%(metadata-filter), 49
- %.~Like\$(metadata-filter), 49
- %.~Like%(metadata-filter), 49
- %.~^Like%(metadata-filter), 49
- %.~^like%(metadata-filter), 49
- %.~eq%(metadata-filter), 49
- %.~in%(metadata-filter), 49
- %.~like\$(metadata-filter), 49
- %.~like%(metadata-filter), 49
- %.~token%(metadata-filter), 49
- %.d%(analytics-dimension), 3
- %.eq%(metadata-filter), 49
- %.f%(analytics-dimension), 3
- %.ge%(metadata-filter), 49
- %.gt%(metadata-filter), 49
- %.ieq%(metadata-filter), 49
- %.in%(metadata-filter), 49
- %.le%(metadata-filter), 49
- %.like\$(metadata-filter), 49
- %.like%(metadata-filter), 49
- %.lt%(metadata-filter), 49
- %.ne%(metadata-filter), 49

- %.teq%(tracked_entity_filter), 54
- %.tew%(tracked_entity_filter), 54
- %.tge%(tracked_entity_filter), 54
- %.tgt%(tracked_entity_filter), 54
- %.tin%(tracked_entity_filter), 54
- %.tle%(tracked_entity_filter), 54
- %.tlike%(tracked_entity_filter), 54
- %.tlt%(tracked_entity_filter), 54
- %.tne%(tracked_entity_filter), 54
- %.token%(metadata-filter), 49
- %.tsw%(tracked_entity_filter), 54

- analytics-dimension, 3
- analytics_dimension
 - (analytics-dimension), 3
- analytics_dimension(), 4, 20, 22, 25, 27

- get_analytics, 4
- get_analytics(), 8, 16, 17, 21, 22, 25–27, 35
- get_analytics_by_level, 5
- get_analytics_outliers, 7
- get_categories (metadata-helpers), 51
- get_category_combos (metadata-helpers), 51
- get_category_option_combos
 - (metadata-helpers), 51
- get_category_option_group_sets
 - (metadata-helpers), 51
- get_category_option_groups
 - (metadata-helpers), 51
- get_category_options
 - (metadata-helpers), 51
- get_complete_data_set_registrations, 8
- get_data_element_group_sets
 - (metadata-helpers), 51
- get_data_element_groups
 - (metadata-helpers), 51
- get_data_elements (metadata-helpers), 51
- get_data_elements_with_category_options, 10

- `get_data_elements_with_category_options()`, 6
- `get_data_sets (metadata-helpers)`, 51
- `get_data_sets()`, 12
- `get_data_sets_by_level`, 11
- `get_data_sets_by_level()`, 8, 9, 17
- `get_data_store_keys`, 12
- `get_data_store_keys()`, 13, 14
- `get_data_store_namespaces`, 13
- `get_data_store_namespaces()`, 12, 14
- `get_data_store_value`, 14
- `get_data_store_value()`, 12, 13
- `get_data_value_audits`, 15
- `get_data_value_sets`, 16
- `get_data_value_sets()`, 8, 9
- `get_dimensions (metadata-helpers)`, 51
- `get_enrollment_analytics`, 20
- `get_enrollment_analytics()`, 21, 22, 26
- `get_enrollment_analytics_aggregate`, 21
- `get_enrollment_analytics_aggregate()`, 27
- `get_enrollments`, 18, 24, 33, 38, 55
- `get_enrollments()`, 20, 21, 23, 24, 33, 53, 55
- `get_event_analytics`, 25
- `get_event_analytics()`, 21, 26, 27
- `get_event_analytics_aggregate`, 26
- `get_event_analytics_aggregate()`, 22
- `get_events`, 19, 23, 33, 38, 55
- `get_events()`, 19, 25, 26, 33, 53, 55
- `get_file_resources`, 28
- `get_geo_features`, 29
- `get_indicator_group_sets (metadata-helpers)`, 51
- `get_indicator_groups (metadata-helpers)`, 51
- `get_indicators (metadata-helpers)`, 51
- `get_metadata`, 30, 53
- `get_metadata()`, 51
- `get_option_group_sets (metadata-helpers)`, 51
- `get_option_groups (metadata-helpers)`, 51
- `get_option_sets (metadata-helpers)`, 51
- `get_options (metadata-helpers)`, 51
- `get_organisation_unit_groups (metadata-helpers)`, 51
- `get_organisation_unit_groupsets (metadata-helpers)`, 51
- `get_organisation_unit_levels (metadata-helpers)`, 51
- `get_organisation_units (metadata-helpers)`, 51
- `get_organisations_by_level`, 31
- `get_organisations_by_level()`, 6, 12
- `get_period_types (metadata-helpers)`, 51
- `get_program_stages (metadata-helpers)`, 51
- `get_programs (metadata-helpers)`, 51
- `get_relationship_types (metadata-helpers)`, 51
- `get_relationships`, 19, 24, 32, 38, 55
- `get_sql_view_data`, 34
- `get_sql_view_data()`, 33, 34
- `get_sql_views`, 33
- `get_sql_views()`, 34, 35
- `get_system_info`, 36
- `get_tracked_entities`, 19, 24, 33, 36, 55
- `get_tracked_entities()`, 19, 23, 24, 33, 53, 55
- `get_tracked_entity_attributes (metadata-helpers)`, 51
- `get_tracked_entity_types (metadata-helpers)`, 51
- `get_user_groups (metadata-helpers)`, 51
- `get_user_profile (metadata-helpers)`, 51
- `get_validation_results`, 39
- `khis_api_version`, 41, 43–48
- `khis_api_version()`, 36
- `khis_base_url`, 42, 42, 44–48
- `khis_cred`, 42, 43, 43, 45–48
- `khis_cred()`, 41, 48
- `khis_cred_clear`, 42–44, 45, 46–48
- `khis_display_name`, 42–45, 46, 47, 48
- `khis_has_cred`, 42–46, 47, 48
- `khis_username`, 42–47, 48
- `khisr-configuration`, 40
- `local_khis_quiet (khisr-configuration)`, 40
- `metadata-filter`, 49
- `metadata-helpers`, 51
- `metadata_filter (metadata-filter)`, 49
- `metadata_filter()`, 28, 30, 34, 37, 54, 55
- `tidyr::unnest_longer()`, 24, 38
- `tidyr::unnest_wider()`, 24, 38

`tracked_entity_filter`, [19](#), [24](#), [33](#), [38](#), [54](#)

`tracked_entity_filter()`, [19](#), [24](#), [37](#), [38](#)

`with_khis_quiet` (`khisr-configuration`),

[40](#)