

# Package ‘khisr’

June 8, 2024

**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) instance. It streamlines data retrieval, empowering researchers, analysts, and healthcare professionals to obtain and utilize data efficiently.

**Version** 1.0.3

**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, 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** 7.3.1

**NeedsCompilation** no

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

**Maintainer** David Kariuki <hello@damurka.com>

**Repository** CRAN

**Date/Publication** 2024-06-08 21:50:13 UTC

## Contents

analytics-dimension . . . . .	2
get_analytics . . . . .	3
get_analytics_by_level . . . . .	4

get_data_elements_with_category_options . . . . .	6
get_data_sets_by_level . . . . .	7
get_metadata . . . . .	8
get_organisations_by_level . . . . .	9
khisr-configuration . . . . .	10
khis_base_url . . . . .	11
khis_cred . . . . .	12
khis_cred_clear . . . . .	13
khis_has_cred . . . . .	13
khis_username . . . . .	14
metadata-filter . . . . .	15
metadata-helpers . . . . .	18

## Index 20

---

analytics-dimension     *Analytics Data Dimensions*

---

### 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): Category 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

*Retrieves Disaggregated Analytics Data from a DHIS2 Server*

---

### 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.
<code>return_type</code>	Optional argument specifying the return format for identifiers. defaults to 'uid'. Choose 'name' for human readable labels.
<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: <ul style="list-style-type: none"> <li>• 0: No output</li> <li>• 1: Show headers</li> <li>• 2: Show headers and bodies</li> <li>• 3: Show headers, bodies, and CURL status message.</li> </ul>
<code>timeout</code>	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 with detailed information

**Examples**

```
# Clinical Breast Examination data elements
# XEX93uLsAm2 = CBE Abnormal
# cXe64Yk0QMY = CBE Normal
element_id <- c('cXe64Yk0QMY', 'XEX93uLsAm2')

# Download data from February 2023 to current date
data <- get_analytics(dx = element_id, pe = 'LAST_MONTH')
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,  
  ...  
)
```

**Arguments**

<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 options that can be passed onto DHIS2 API.

**Details**

- 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.

**Value**

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

**See Also**

- [get\\_organisation\\_unit\\_levels\(\)](#) for getting the organisations units
- [get\\_data\\_elements\\_with\\_category\\_options\(\)](#) for retrieving the data elements

**Examples**

```
# Clinical Breast Examination data elements
# XEX93uLsAm2 = CBE Abnormal
# cXe64Yk0QMY = CBE Normal
element_id = c('cXe64Yk0QMY', 'XEX93uLsAm2')

# Download data from February 2023 to current date
data <- get_analytics_by_level(element_ids = element_id,
                             start_date = '2023-02-01')

data
```

---

```
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)
```

**Arguments**

`element_ids`     The data element identifiers whose details being retrieved

**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('htFuvGJRW1X')
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,  
  ...  
)
```

### Arguments

<code>dataset_ids</code>	Required vector of data sets IDs for which to retrieve data. Required.
<code>start_date</code>	Optional 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>	Required desired organisation level of data (default: level 1) .
<code>org_ids</code>	Optional list of organization units IDs to be filtered.
<code>...</code>	Other options that can be passed onto DHIS2 API.

### 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 MoH 745 Cancer Screening Program Monthly Summary Form
dataset_id = c('WWh5hbCmvND')

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

data
```

---

get\_metadata

*Get Metadata from a DHIS2 Instance*


---

**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
)
```

**Arguments**

endpoint	The DHIS2 API endpoint for the metadata of interest (e.g., dataElements, organisationUnits endpoints).
...	One or more <code>metadata_filter()</code> 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: <ul style="list-style-type: none"> <li>• 0: No output</li> <li>• 1: Show headers</li> <li>• 2: Show headers and bodies</li> <li>• 3: Show headers, bodies, and CURL status message.</li> </ul>
timeout	Maximum number of seconds to wait for the DHIS2 API response.

**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 'WWh5hbCmvND'
get_metadata('dataSets',
            fields = 'id,name,organisationUnits[id,name,path]',
            id %.eq% 'WWh5hbCmvND')

# Get data elements filtered by dataElementGroups id
get_metadata('dataElements',
            dataElementGroups.id %.eq% 'IXd7DXxZqzL',
            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)
```

## 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.

## Value

A tibble containing the organisation units and their parent units up to the specified level.

## Examples

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

---

khisr-configuration    *khisr Configuration*

---

## Description

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

## 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_quite` to run small bit of code silently.

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

## Examples

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

# suppress messages for a small amount of code
with_khis_quiet(
  khis_cred(username = 'username',
```

```

        password = 'password',
        base_url = 'https://dhis2-instance/api')
    )

## End(Not run)

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

# suppress messages for a in a specific scope
local_khis_quiet()

# no message
khis_cred(username = 'username',
          password = 'password',
          base_url = 'https://dhis2-instance/api')

# clear credentials
khis_cred_clear()

## End(Not run)

```

---

khis_base_url	<i>Produces the Configured DHIS2 API Base URI</i>
---------------	---

---

### **Description**

Produces the Configured DHIS2 API Base URI

### **Usage**

```
khis_base_url()
```

### **Value**

the DHIS2 base URI

### **Examples**

```

## Not run:
# Set the credentials
khis_cred(username = 'DHIS2 username',
          password = 'DHIS2 password',
          base_url = 'https://dhis2-instance/api')

# View the DHIS2 instance API expect 'https://dhis2-instance/api'
khis_base_url()

```

```
# Clear credentials
khis_cred_clear()

## End(Not run)
```

---

**khis\_cred***Sets DHIS2 Credentials*

---

### Description

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

### Usage

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

### Arguments

<code>config_path</code>	An optional path to a configuration file containing username and password. This is considered more secure than providing credentials directly in code.
<code>username</code>	The DHIS2 username. Only required if configuration file not provided.
<code>password</code>	The DHIS2 password. Only required if configuration file not provided.
<code>base_url</code>	The base URL of the DHIS2 instance. Only required if configuration file not provided.

### Details

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

### Value

No return value

### See Also

Other credential functions: [khis\\_cred\\_clear\(\)](#), [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 credentials directly (less secure)
khis_cred(username = "your_username",
           password = "your_password",
           base_url='https://dhis2-instance/api')

## End(Not run)
```

---

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

---

**Description**

Clear the Credentials from Memory

**Usage**

```
khis_cred_clear()
```

**Value**

No return value

**See Also**

Other credential functions: [khis\\_cred\(\)](#), [khis\\_has\\_cred\(\)](#), [khis\\_username\(\)](#)

**Examples**

```
#khis_cred_clear()
```

---

khis_has_cred	<i>Are There Credentials on Hand?</i>
---------------	---------------------------------------

---

**Description**

Are There Credentials on Hand?

**Usage**

```
khis_has_cred()
```

**Value**

a boolean value indicating if the credentials are available

**See Also**

Other credential functions: [khis\\_cred\(\)](#), [khis\\_cred\\_clear\(\)](#), [khis\\_username\(\)](#)

**Examples**

```
## Not run:
# Set the credentials
khis_cred(username = 'DHIS2 username',
           password = 'DHIS2 password',
           base_url='https://dhis2-instance/api')

# Check if credentials available. Expect TRUE
khis_has_cred()

# Clear credentials
khis_cred_clear()

# Check if credentials available. Expect FALSE
khis_has_cred()

## End(Not run)
```

---

khis\_username

*Produces the Configured Username*

---

**Description**

Produces the Configured Username

**Usage**

```
khis_username()
```

**Value**

the username of the user credentials

**See Also**

Other credential functions: [khis\\_cred\(\)](#), [khis\\_cred\\_clear\(\)](#), [khis\\_has\\_cred\(\)](#)

**Examples**

```
## Not run:
# Set the credentials
khis_cred(username = 'DHIS2 username',
           password = 'DHIS2 password',
           base_url = 'https://dhis2-instance/api')

# 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	description

### 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)
```

**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(...)
```

### 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.

### Value

A tibble containing the DHIS2 metadata response.

### 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('VR7vdS7P0Gb', 'gQro1y7Rsbq'))  
  
# Get datasets by name with the word 'MOH 705'  
get_data_sets(name %.like% 'MOH 705')
```

# Index

## \* credential functions

- khis\_cred, 12
- khis\_cred\_clear, 13
- khis\_has\_cred, 13
- khis\_username, 14
- %.Like\$(metadata-filter), 15
- %.Like% (metadata-filter), 15
- %.^Like% (metadata-filter), 15
- %.^like% (metadata-filter), 15
- %.~Like\$(metadata-filter), 15
- %.~Like% (metadata-filter), 15
- %.~^Like% (metadata-filter), 15
- %.~^like% (metadata-filter), 15
- %.~eq% (metadata-filter), 15
- %.~in% (metadata-filter), 15
- %.~like\$(metadata-filter), 15
- %.~like% (metadata-filter), 15
- %.~token% (metadata-filter), 15
- %.d% (analytics-dimension), 2
- %.eq% (metadata-filter), 15
- %.f% (analytics-dimension), 2
- %.ge% (metadata-filter), 15
- %.gt% (metadata-filter), 15
- %.ieq% (metadata-filter), 15
- %.in% (metadata-filter), 15
- %.le% (metadata-filter), 15
- %.like\$(metadata-filter), 15
- %.like% (metadata-filter), 15
- %.lt% (metadata-filter), 15
- %.ne% (metadata-filter), 15
- %.token% (metadata-filter), 15
- analytics-dimension, 2
- analytics\_dimension
  - (analytics-dimension), 2
- analytics\_dimension(), 4
- get\_analytics, 3
- get\_analytics\_by\_level, 4
- get\_categories (metadata-helpers), 18

- get\_category\_combos (metadata-helpers), 18
- get\_category\_option\_combos (metadata-helpers), 18
- get\_category\_option\_group\_sets (metadata-helpers), 18
- get\_category\_option\_groups (metadata-helpers), 18
- get\_category\_options (metadata-helpers), 18
- get\_data\_element\_group\_sets (metadata-helpers), 18
- get\_data\_element\_groups (metadata-helpers), 18
- get\_data\_elements (metadata-helpers), 18
- get\_data\_elements\_with\_category\_options, 6
- get\_data\_elements\_with\_category\_options(), 5
- get\_data\_sets (metadata-helpers), 18
- get\_data\_sets(), 7
- get\_data\_sets\_by\_level, 7
- get\_dimensions (metadata-helpers), 18
- get\_indicator\_group\_sets (metadata-helpers), 18
- get\_indicator\_groups (metadata-helpers), 18
- get\_indicators (metadata-helpers), 18
- get\_metadata, 8, 19
- get\_metadata(), 18
- get\_option\_group\_sets (metadata-helpers), 18
- get\_option\_groups (metadata-helpers), 18
- get\_option\_sets (metadata-helpers), 18
- get\_options (metadata-helpers), 18
- get\_organisation\_unit\_groups (metadata-helpers), 18
- get\_organisation\_unit\_groupsets (metadata-helpers), 18

get\_organisation\_unit\_levels  
    (metadata-helpers), 18  
get\_organisation\_unit\_levels(), 5  
get\_organisation\_units  
    (metadata-helpers), 18  
get\_organisations\_by\_level, 9  
get\_organisations\_by\_level(), 7  
get\_period\_types (metadata-helpers), 18  
get\_user\_groups (metadata-helpers), 18  
  
khis\_base\_url, 11  
khis\_cred, 12, 13, 14  
khis\_cred\_clear, 12, 13, 14  
khis\_has\_cred, 12, 13, 13, 14  
khis\_username, 12–14, 14  
khisr-configuration, 10  
  
local\_khis\_quiet (khisr-configuration),  
    10  
  
metadata-filter, 15  
metadata-helpers, 18  
metadata\_filter (metadata-filter), 15  
metadata\_filter(), 8  
  
with\_khis\_quiet (khisr-configuration),  
    10