

# Package ‘cloudos’

October 12, 2022

**Title** R Client Library for CloudOS

**Version** 0.3.0

**Description** The 'CloudOS' client library for R makes it easy to interact with CloudOS <<https://cloudos.lifebit.ai/>> in the R environment for analysis.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.1.2

**Imports** httr, utils, methods, dplyr (>= 1.0.0), tidy, jsonlite, rappdirs, ggplot2, tibble

**Suggests** httpptest, kableExtra, knitr, lintr, rmarkdown, testthat

**URL** <https://github.com/lifebit-ai/cloudos>

**BugReports** <https://github.com/lifebit-ai/cloudos/issues>

**NeedsCompilation** no

**Author** Sangram Keshari Sahu [aut, cre],  
Ilya Levantis [aut],  
Henry Stewart [aut],  
Lifebit Biotech Ltd. [cph]

**Maintainer** Sangram Keshari Sahu <[sangram@lifebit.ai](mailto:sangram@lifebit.ai)>

**Repository** CRAN

**Date/Publication** 2022-02-09 10:20:02 UTC

## R topics documented:

.get_search_json . . . . .	2
.v1_query_to_v2 . . . . .	2
cb_apply_query . . . . .	3
cb_create_cohort . . . . .	4
cb_get_genotypic_table . . . . .	4
cb_get_participants_table . . . . .	5
cb_get_participants_table_long . . . . .	6

cb_get_phenotype_metadata . . . . .	6
cb_get_phenotype_statistics . . . . .	7
cb_list_cohorts . . . . .	8
cb_load_cohort . . . . .	8
cb_participant_count . . . . .	9
cb_plot_filters . . . . .	9
cb_search_phenotypes . . . . .	10
cb_set_columns . . . . .	11
cloudos_configure . . . . .	11
cloudos_whoami . . . . .	12
cohort-class . . . . .	12
phenotype . . . . .	13

## Index 14

---

.get_search_json	<i>only used for v1 endpoint - creates v1 search json using the v2 style query</i>
------------------	--

---

### Description

only used for v1 endpoint - creates v1 search json using the v2 style query

### Usage

```
.get_search_json(my_cohort)
```

### Arguments

my_cohort	A cohort object
-----------	-----------------

---

.v1_query_to_v2	<i>Convert a v1 style query (moreFields) to v2 style (query). v2 queries are a superset of v1 queries. A list of v1 phenotype queries are equivalent to a set of nested v2 AND operators containing those phenotypes. This function builds the nested AND query from the flat list of v1 phenotypes.</i>
-----------------	--

---

### Description

Convert a v1 style query (moreFields) to v2 style (query). v2 queries are a superset of v1 queries. A list of v1 phenotype queries are equivalent to a set of nested v2 AND operators containing those phenotypes. This function builds the nested AND query from the flat list of v1 phenotypes.

### Usage

```
.v1_query_to_v2(cohort_more_fields)
```

**Arguments**

cohort\_more\_fields  
 query information ('moreFields') from .get\_cohort\_info(cohort\_id, cb\_version="v1)

---

cb_apply_query	<i>Apply a query to a cohort</i>
----------------	----------------------------------

---

**Description**

Updates a cohort by applying a new query.

**Usage**

```
cb_apply_query(cohort, query, keep_query = TRUE)
```

**Arguments**

cohort	A cohort object. (Required) See constructor function <a href="#">cb_create_cohort</a> or <a href="#">cb_load_cohort</a>
query	A phenotype query defined using the <a href="#">phenotype</a> function and logic operators (see example below)
keep_query	If True, combines the newly supplied query with the pre-existing query. Otherwise, pre-existing query is overwritten. (Default: TRUE)

**Value**

The updated cohort object.

**Examples**

```
## Not run:
A <- phenotype(id = 13, from = "2016-01-21", to = "2017-02-13")
B <- phenotype(id = 4, value = "Cancer")

A_not_B <- A & !B

my_cohort <- cb_load_cohort(cohort_id = "612f37a57673ed0ddeaf1333", cb_version = "v2")

my_cohort <- cb_apply_query(my_cohort, query = A_not_B, keep_query = F)

## End(Not run)
```

---

cb\_create\_cohort      *Create Cohort*

---

### Description

Creates a new Cohort

### Usage

```
cb_create_cohort(cohort_name, cohort_desc, filters = "", cb_version = "v2")
```

### Arguments

cohort_name	New cohort name to be created. (Required)
cohort_desc	New cohort description to be created. (Optional)
filters	WIP - details will be added.
cb_version	cohort browser version. ["v1"   "v2"] (Optional) Default - "v2"

### Value

A [cohort](#) object.

### See Also

[cb\\_load\\_cohort](#) for loading a available cohort.

### Examples

```
## Not run:
my_cohort <- cb_create_cohort(cohort_name = "Cohort-R",
                             cohort_desc = "This cohort is for testing purpose, created from R.")

## End(Not run)
```

---

cb\_get\_genotypic\_table      *Get genotypic table*

---

### Description

Get Genotypic table in a dataframe. Optionally genotypic filters can be applied as well.

### Usage

```
cb_get_genotypic_table(cohort, size = 10, geno_filters_query)
```

**Arguments**

cohort	A cohort object. (Required) See constructor functions <a href="#">cb_create_cohort</a> or <a href="#">cb_load_cohort</a>
size	Number of entries from database. (Optional) Default - 10 (Optional)
geno_filters_query	Genotypic filter query (Optional)

**Value**

A dataframe.

---

cb\_get\_participants\_table  
*Get participant data table*

---

**Description**

Get participant data table in a dataframe.

**Usage**

```
cb_get_participants_table(cohort, cols, page_number = "all", page_size = 5000)
```

**Arguments**

cohort	A cohort object. (Required) See constructor functions <a href="#">cb_create_cohort</a> or <a href="#">cb_load_cohort</a>
cols	Vector of phenotype IDs to fetch as columns in the dataframe. If omitted, columns saved in the cohort are fetched.
page_number	Number of page as integer or 'all' to fetch all data. (Optional) Default - 'all'
page_size	Number of entries in a page. (Optional) Default - 5000

**Value**

A dataframe.

---

 cb\_get\_participants\_table\_long

*Get longform participant data table*


---

### Description

Get participant data table in a longform dataframe.

### Usage

```
cb_get_participants_table_long(
  cohort,
  cols,
  broadcast = TRUE,
  page_number = 0,
  page_size = 100
)
```

### Arguments

cohort	A cohort object. (Required) See constructor functions <a href="#">cb_create_cohort</a> or <a href="#">cb_load_cohort</a>
cols	Vector of phenotype IDs to fetch as columns in the dataframe. If omitted, columns saved in the cohort are fetched.
broadcast	Whether to broadcast single value phenotypes across rows. (Optional) Can be TRUE, FALSE or a vector of phenotype IDs to specify which phenotypes to broadcast. Default - TRUE
page_number	Number of page (can be 'all' to fetch all data) . (Optional) Default - 0
page_size	Number of entries in a page. (Optional) Default - 10

### Value

A tibble.

---

 cb\_get\_phenotype\_metadata

*Phenotype metadata*


---

### Description

Get the metadata of a phenotype in the cohort browser

### Usage

```
cb_get_phenotype_metadata(pheno_id, cb_version = "v2")
```

**Arguments**

pheno_id	A phenotype ID. (Required)
cb_version	cohort browser version. (Default: "v2") [ "v1"   "v2" ]

**Value**

A data frame.

---

cb\_get\_phenotype\_statistics

*Get distribution of a phenotype in a cohort*

---

**Description**

Retrieve a data frame containing the distribution data for a specific phenotype within a cohort.

**Usage**

```
cb_get_phenotype_statistics(
  cohort,
  pheno_id,
  max_depth = Inf,
  page_number = "all",
  page_size = 1000
)
```

**Arguments**

cohort	A cohort object. (Required) See constructor function <a href="#">cb_create_cohort</a> or <a href="#">cb_load_cohort</a>
pheno_id	A phenotype ID. (Required)
max_depth	The maximum depth to descend in a 'nested list' phenotype. (Default: Inf)
page_number	For internal use.
page_size	For internal use.

**Value**

A data frame holding distribution data.

---

cb_list_cohorts	<i>List cohorts</i>
-----------------	---------------------

---

**Description**

Extracts the data frame with limited cohort data columns.

**Usage**

```
cb_list_cohorts(size = 10, cb_version = "v2")
```

**Arguments**

size	Number of cohort entries from database. (Optional) Default - 10
cb_version	cohort browser version. ["v1"   "v2"] (Optional) Default - "v2"

**Value**

A data frame with available cohorts.

**Examples**

```
## Not run:
cohorts_list()

## End(Not run)
```

---

cb_load_cohort	<i>Get cohort information</i>
----------------	-------------------------------

---

**Description**

Get all the details about a cohort including applied query.

**Usage**

```
cb_load_cohort(cohort_id, cb_version = "v2")
```

**Arguments**

cohort_id	Cohort id (Required)
cb_version	cohort browser version (Optional) [ "v1"   "v2" ]

**Value**

A [cohort](#) object.



**See Also**

[cb\\_create\\_cohort](#) for creating a new cohort.

---

*cb\_participant\_count*    *Participant Count*

---

**Description**

Returns the number of participants in a cohort if the supplied query were to be applied.

**Usage**

```
cb_participant_count(cohort, query = list(), keep_query = TRUE)
```

**Arguments**

- cohort            A cohort object. (Required) See constructor function [cb\\_create\\_cohort](#) or [cb\\_load\\_cohort](#)
- query            A phenotype query defined using the [codephenotype](#) function and logic operators (see example below)
- keep\_query       Apply newly specified query on top of existing query (Default: TRUE)

**Value**

A list with count of participants in the cohort and the total no. of participants in the dataset.

---

*cb\_plot\_filters*            *Plot filters*

---

**Description**

Get a list of ggplot objects, each plot having one filter.

**Usage**

```
cb_plot_filters(cohort)
```

**Arguments**

- cohort            A cohort object. (Required) See constructor function [cb\\_create\\_cohort](#) or [cb\\_load\\_cohort](#)

**Value**

A list of ggplot objects

### Examples

```
## Not run:
my_cohort <- cb_load_cohort(cohort_id = "5f9af3793dd2dc6091cd17cd")
plot_list <- cb_plot_filters(cohort = my_cohort)

plot_list[[1]]

library(ggpubr)
ggpubr::ggarrange(plotlist = plot_list)

## End(Not run)
```

---

cb\_search\_phenotypes    *Search available phenotypes*

---

### Description

Search for phenotypes in the Cohort Browser that match your term and return a tibble containing the metadata information for each matching phenotype. Use 'term = ""' to return all phenotypes.

### Usage

```
cb_search_phenotypes(term, cb_version = "v2")
```

### Arguments

term	A term to search. (Required)
cb_version	cohort browser version (Optional) [ "v1"   "v2" ]

### Value

A tibble with phenotype metadata

### Examples

```
## Not run:
cancer_phenos <- cb_search_phenotypes(term = "cancer")

all_phenos <- cb_search_phenotypes(term = "")

## End(Not run)
```

---

cb_set_columns	<i>Set the columns in a cohort</i>
----------------	------------------------------------

---

**Description**

Updates a cohort by applying a new query.

**Usage**

```
cb_set_columns(cohort, column_ids, keep_columns = TRUE)
```

**Arguments**

cohort	A cohort object. (Required) See constructor function <a href="#">cb_create_cohort</a> or <a href="#">cb_load_cohort</a>
column_ids	Vector of phenotype IDs to be added as columns in the participant table.
keep_columns	If True, pre-existing columns are retained and newly supplied columns are added. Otherwise, pre-existing columns are overwritten. (Default: TRUE)

**Value**

The updated cohort object.

**Examples**

```
## Not run:  
my_cohort <- cb_load_cohort(cohort_id = "612f37a57673ed0ddeaf1333", cb_version = "v2")  
  
my_cohort <- cb_set_columns(my_cohort, c(1, 99, 38), keep_columns = F)  
  
## End(Not run)
```

---

cloudos_configure	<i>Configure cloudos</i>
-------------------	--------------------------

---

**Description**

On a system for the first time the cloudos configuration needed to be done. This function can help do that.

**Usage**

```
cloudos_configure(base_url, token, team_id)
```

**Arguments**

base_url	Base URL for cloudos
token	API key or token
team_id	team/workspace ID

**Value**

None

---

cloudos_whoami	<i>whoami</i>
----------------	---------------

---

**Description**

To check the current configuration

**Usage**

cloudos\_whoami()

**Value**

None

---

cohort-class	<i>cohort class</i>
--------------	---------------------

---

**Description**

This class creates a cohort object, which holds the information related to a cohort: cohort ID, name, description, query, table columns. This class is used in functions which carry out operations related to specific cohorts. A cohort class object can be created using constructor functions [cb\\_create\\_cohort](#) or [cb\\_load\\_cohort](#).

**Slots**

id cohort ID.  
 name cohort name.  
 desc cohort description.  
 phenotype\_filters phenotypes displayed in the cohort overview.  
 query applied query.  
 query\_phenotype\_ids IDs of phenotypes used in the query.  
 columns All the columns.  
 num\_participants number of participants in the cohort.  
 cb\_version cohort browser version.

---

phenotype	<i>Define a phenotype</i>
-----------	---------------------------

---

**Description**

Defines a single phenotype

**Usage**

```
phenotype(id, value, from, to, instance = "0")
```

**Arguments**

id	A single phenotype id. Possible phenotyoes can be explored using the <a href="#">codecb_search_phenotypes</a> function
value	The categorical value of the phenotype id defined
from	For continuous phenotypes, the lower bound of the desired value range
to	For continuous phenotypes, the upper bound of the desired value phenotype
instance	The instance number of the phenotype, default 0

**Value**

A single phenotypes definition that cam be combined using &,| and ! operators

**Examples**

```
## Not run:  
continuous_phenotype <- phenotype(id = 13, from = "2016-01-21", to = "2017-02-13")  
categorical_phenotype <- phenotype(id = 4, value = "Cancer")  
  
## End(Not run)
```

# Index

.get\_search\_json, 2  
.v1\_query\_to\_v2, 2

cb\_apply\_query, 3  
cb\_create\_cohort, 3, 4, 5–7, 9, 11, 12  
cb\_get\_genotypic\_table, 4  
cb\_get\_participants\_table, 5  
cb\_get\_participants\_table\_long, 6  
cb\_get\_phenotype\_metadata, 6  
cb\_get\_phenotype\_statistics, 7  
cb\_list\_cohorts, 8  
cb\_load\_cohort, 3–7, 8, 9, 11, 12  
cb\_participant\_count, 9  
cb\_plot\_filters, 9  
cb\_search\_phenotypes, 10, 13  
cb\_set\_columns, 11  
cloudos\_configure, 11  
cloudos\_whoami, 12  
cohort, 4, 8  
cohort-class, 12

phenotype, 3, 9, 13