Package ‘queryup’

October 13, 2022

Type Package
Title Query the 'UniProtKB' REST API
Version 1.0.0
Maintainer Guillaume Voisinne <voisinne@gmail.com>
Description Retrieve protein information from the 'UniProtKB' REST API (see <https://www.uniprot.org/help/api_queries>).
License GPL-3
URL https://github.com/VoisinneG/queryup
BugReports https://github.com/VoisinneG/queryup/issues
Depends R (>= 2.10)
Imports jsonlite, RCurl, utils
Suggests covr, knitr, rmarkdown, testthat (>= 3.0.0)
Config/testthat/edition 3
Encoding UTF-8
LazyData true
RoxygenNote 7.1.2
VignetteBuilder knitr
NeedsCompilation no
Author Guillaume Voisinne [aut, cre] (<https://orcid.org/0000-0003-3922-2121>)
Repository CRAN
Date/Publication 2022-09-12 07:40:04 UTC

R topics documented:

  build_query_url .................................................. 2
  clean_query ...................................................... 2
  get_uniprot_data .................................................. 3
  parse_messages .................................................... 4
  query_fields ....................................................... 4
**build_query_url**

Accessory function used to build the query url

**Description**

Accessory function used to build the query url

**Usage**

```r
build_query_url(
  query = NULL,
  columns = c("accession", "id", "gene_names", "organism_name", "reviewed"),
  format = "json"
)
```

**Arguments**

- **query**
  - list of keys corresponding to UniProt's query fields. For example: `list("gene_exact" = c("Pik3r1", "Pik3r2"), "organism" = c("10090", "9606"), "reviewed" = "yes")`

- **columns**
  - names of UniProt data columns to retrieve.

- **format**
  - format of the response provided by the UniProt API

**Value**

- the query url

---

**clean_query**

Accessory function removing invalid values from a query

**Description**

Accessory function removing invalid values from a query

**Usage**

```r
clean_query(query, df)
```
get_uniprot_data

Arguments

query list of keys corresponding to uniprot's query fields. For example: list("gene_exact" = c("Pik3r1", "Pik3r2"), "organism" = c("10090", "9606"), "reviewed" = "yes")

df data.frame with invalid values (in column "value") and corresponding query field (in column "field").

Value

the input query without the invalid values

get_uniprot_data Retrieve data from UniProt using UniProt’s REST API

Description

Retrieve data from UniProt using UniProt’s REST API

Usage

get_uniprot_data(
  query = NULL,
  columns = c("accession", "id", "gene_names", "organism_id", "reviewed"),
  print_url = FALSE,
  print_uniprot_messages = FALSE
)

Arguments

query list of keys corresponding to UniProt’s query fields. For example: list("gene_exact" = c("Pik3r1", "Pik3r2"), "organism" = c("10090", "9606"), "reviewed" = "yes"). See ‘query_fields’ for available query fields.

columns names of UniProt data columns to retrieve. Examples include "accession", "id", "gene_names", "keyword", "sequence". See ‘return_fields’ for available return fields.

print_url Boolean. Prints the complete url used for the query.

print_uniprot_messages Boolean. Prints the raw error message returned by UniProt.

Value

a list with the following items:

url the query url

messages messages returned by the REST API

content a data.frame containing the query results
Examples

```r
# Getting gene names, keywords and protein sequences for a set of UniProt IDs.
ids <- c("P22682", "P47941")
cols <- c("accession", "id", "gene_names", "keyword", "sequence")
query = list("accession_id" = ids)
df <- get_uniprot_data(query = query, columns = cols)$content
df
```

---

`parse_messages`  
Accessory function retrieving invalid values from messages returned by the UniProt API.

**Description**

Accessory function retrieving invalid values from messages returned by the UniProt API.

**Usage**

`parse_messages(messages)`

**Arguments**

- `messages` character string containing the error messages returned by UniProt API

**Value**

A data.frame with invalid values (in column "value") and corresponding query field (in column "field"). NULL if no invalid values are identified.

---

`query_fields`  
Available query fields.

**Description**

Query fields that can be used to generate queries using 'queryup' along with associated examples and description.

**Usage**

`query_fields`

**Format**

A data frame with 44 rows and 3 variables:

- `field` Name of the query field
- `example` Example query (as appearing in the query url)
- `description` Description of the example query
query_uniprot

Retrieve data from UniProt using UniProt's REST API.

Description

Retrieve data from UniProt using UniProt's REST API. To avoid non-responsive queries, they are split into smaller queries with at most max_keys items per query field. Not that it works only with queries where items within query fields are collapsed with '+OR+' and different query fields are collapsed with '+AND+' (see query_uniprot())

Usage

query_uniprot(
  query = NULL,
  columns = c("accession", "id", "gene_names", "organism_id", "reviewed"),
  print_url = FALSE,
  print_uniprot_messages = FALSE,
  max_keys = 300,
  updateProgress = NULL,
  show_progress = TRUE
)

Arguments

query list of keys corresponding to UniProt's query fields. For example : query = list("gene_exact" = c("Pik3r1", "Pik3r2"), "organism_id" = c("10090", "9606"), "reviewed" = "true"). See 'query_fields' for available query fields.

columns names of UniProt data columns to retrieve. Examples include "accession", "id", "genes", "keywords", "sequence". See 'return_fields' for available return fields.

print_url logical. Prints the complete url used for the query.

print_uniprot_messages Boolean. Prints the raw error message returned by UniProt.

max_keys maximum number of field items submitted

updateProgress used to display progress in shiny apps

show_progress Show progress bar

Value

a data.frame
Examples

# Get the UniProt entries of all proteins encoded by gene Pik3r1
query <- list("gene_exact" = "Pik3r1")
df <- query_uniprot(query = query)
head(df)

return_fields

Available return fields ("columns").

Description

Return fields that can be retrieved using 'queryup' along with their label (column "Label") as appearing in the retrieved data.frame.

Usage

return_fields

Format

A data frame with 287 rows and 2 variables:

field Name of the returned field
label Label of the corresponding column in the retrieved data.frame

Source

https://www.uniprot.org/help/return_fields

uniprot_entries

Information for 1000 UniProt entries from the organism Mus musculus

Description

Entry names and other attributes of 1000 UniProt entries in Mus musculus.

Usage

uniprot_entries
uniprot_entries

Format
A data frame with 1000 rows and 5 variables:

- **Entry**: UniProt entry accession id
- **Entry Name**: UniProt entry name
- **Gene Names**: Gene names
- **Organism (ID)**: Taxon ID
- **Reviewed**: Swiss-Prot review status

Source

https://www.uniprot.org/
Index

* datasets
  query_fields, 4
  return_fields, 6
  uniprot_entries, 6

build_query_url, 2

clean_query, 2

get_uniprot_data, 3

parse_messages, 4

query_fields, 4
query_uniprot, 5

return_fields, 6

uniprot_entries, 6