

Package ‘queryup’

June 5, 2023

Type Package

Title Query the 'UniProtKB' REST API

Version 1.0.5

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 httr, jsonlite, utils

Suggests covr, knitr, xml2, 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 2023-06-05 18:30:02 UTC

R topics documented:

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

Index	7
--------------	----------

build_query_url	<i>Accessory function used to build the query url</i>
-----------------	---

Description

Accessory function used to build the query url

Usage

```
build_query_url(
  query = NULL,
  base_url = "https://rest.uniprot.org/uniprotkb/",
  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")
base_url	The base url for the UniProt REST API
columns	names of UniProt data columns to retrieve.
format	format of the response provided by the UniProt API

Value

the query url

clean_query	<i>Accessory function removing invalid values from a query</i>
-------------	--

Description

Accessory function removing invalid values from a query

Usage

```
clean_query(query, df)
```

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	<i>Retrieve data from UniProt using UniProt's REST API</i>
------------------	--

Description

Retrieve data from UniProt using UniProt's REST API

Usage

```
get_uniprot_data(
  query = NULL,
  base_url = "https://rest.uniprot.org/uniprotkb/",
  columns = c("accession", "id", "gene_names", "organism_id", "reviewed")
)
```

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.
base_url	The base url for the UniProt REST API
columns	names of UniProt data columns to retrieve. Examples include "accession", "id", "gene_names", "keyword", "sequence". See 'return_fields' for available return fields.

Value

a list with the following items :

url the query url

status the http status code for the request

messages messages returned by the REST API

content a data.frame containing the query results

Examples

```
#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	<i>Accessory function retrieving invalid values from messages returned by the UniProt API.</i>
----------------	--

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	<i>Available query fields.</i>
--------------	--------------------------------

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

Source

<https://www.uniprot.org/help/query-fields>

query_uniprot	<i>Retrieve data from UniProt using UniProt's REST API.</i>
---------------	---

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. Note 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,
  base_url = "https://rest.uniprot.org/uniprotkb/",
  columns = c("accession", "id", "gene_names", "organism_id", "reviewed"),
  max_keys = 200,
  updateProgress = NULL,
  show_progress = TRUE
)
```

Arguments

query	list of keys corresponding to UniProt's query fields. For example : <code>query = list("gene_exact" = c("Pik3r1", "Pik3r2"), "organism_id" = c("10090", "9606"), "reviewed" = "true")</code> . See 'query_fields' for available query fields.
base_url	The base url for the UniProt REST API
columns	names of UniProt data columns to retrieve. Examples include "accession", "id", "genes", "keywords", "sequence". See 'return_fields' for available return fields.
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
ids <- c("P22682", "P47941")
query = list("accession_id" = ids)
df <- query_uniprot(query = query)
head(df)
```

return_fields	<i>Available return fields ("columns").</i>
---------------	---

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	<i>Information for 1000 UniProt entries from the organism Mus musculus</i>
-----------------	--

Description

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

Usage

```
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