

Package ‘reutils’

September 3, 2016

Type Package

Title Talk to the NCBI EUtils

Version 0.2.3

Date 2016-09-03

Author Gerhard Schöfl <gerhard.schofl@gmail.com>

Maintainer Gerhard Schöfl <gerhard.schofl@gmail.com>

Description An interface to NCBI databases such as PubMed, GenBank, or GEO powered by the Entrez Programming Utilities (EUtils). The nine EUtils provide programmatic access to the NCBI Entrez query and database system for searching and retrieving biological data.

Depends R (>= 3.2.0)

Imports assertthat, jsonlite, methods, RCurl, stats, XML, tibble (>= 1.2)

Suggests testthat (>= 0.8)

License MIT + file LICENSE

Encoding UTF-8

URL <https://github.com/gschofl/reutils>

BugReports <https://github.com/gschofl/reutils/issues>

Collate 'utils.R' 'util-error.R' 'util.R' 'ecitmatch.R'
'parse-params.R' 'efetch.R' 'egquery.R' 'einfo.R' 'elink.R'
'epost.R' 'esearch.R' 'espell.R' 'esummary.R' 'parse-docsum.R'
'reutils-package.R' 'zzz.R'

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2016-09-03 16:39:17

R topics documented:

reutils-package	2
content	5
database	7
ecitmatch	8
efetch	9
egquery	11
einfo	12
elink	13
epost	15
esearch	16
espell	18
esummary	18
getError	20
getUrl	21
linkset	22
querykey	23
retmode	24
rettype	25
uid	26
webenv	27
[,efetch,character,missing-method	28
[,einfo,ANY,missing-method	29
[,elink,ANY,missing-method	30
[,esearch,numeric,missing-method	30
[,esummary,character,ANY-method	31
Index	33

reutils-package *Talk to the NCBI EUtils*

Description

An interface to NCBI databases such as PubMed, GenBank, or GEO powered by the Entrez Programming Utilities (EUtils). The nine EUtils provide programmatic access to the NCBI Entrez query and database system for searching and retrieving biological data.

Details

With nine **Entrez Programming Utilities**, NCBI provides a programmatic interface to the Entrez query and database system for searching and retrieving requested data

Each of these tools corresponds to an R function in the `reutils` package described below.

The output returned by the EUtils is typically in XML format. To gain access to this output you have several options:

1. Use the `content(as = "xml")` method to extract the output as an `XMLInternalDocument` object and process it further using the facilities provided by the XML package.
2. Use the `content(as = "parsed")` method to extract the output into `data.frames`. Note that this is currently only implemented for `docsums` returned by `esummary`, `uילים` returned by `esearch`, and the output returned by `einfo`.
3. Access specific nodes in the XML tree using XPath expressions with the reference class methods `#xmlValue`, `#xmlAttr`, or `#xmlName` built into `eutil` objects.

The Entrez Programming Utilities can also generate output in other formats, such as plain-text Fasta or GenBank files for sequence databases, or the MedLine format for the literature database. The type of output is generally controlled by setting the `retmode` and `rettype` arguments when calling a `EUtil`. Please check the relevant [usage guidelines](#) when using these services. Note that Entrez server requests are subject to frequency limits.

Main functions

- `esearch`: Search and retrieve primary UIDs for use with `esummary`, `elink`, or `efetch`. `esearch` additionally returns term translations and optionally stores results for future use in the user's Web Environment.
- `esummary`: Retrieve document summaries from a list of primary UIDs (Provided as a character vector or as an `esearch` object).
- `egquery`: Provides Entrez database counts in XML for a single search term using a Global Query.
- `einfo`: Retrieve field names, term counts, last update, and available updates for each database.
- `efetch`: Retrieve data records in a specified format corresponding to a list of primary UIDs or from the user's Web Environment in the Entrez History server.
- `elink`: Returns a list of UIDs (and relevancy scores) from a target database that are related to a list of UIDs in the same database or in another Entrez database.
- `epost`: Uploads primary UIDs to the user's Web Environment on the Entrez history server for subsequent use with `esummary`, `elink`, or `efetch`.
- `espell`: Provide spelling suggestions.
- `ecitmatch`: Retrieves PubMed IDs (PMIDs) that correspond to a set of input citation strings
- `content`: Extract the content of a request from the `eutil` object returned by any of the above functions.

Package options

`reutils` uses three [options](#) to configure behaviour:

- `reutils.email`: NCBI requires that a user of their API provides an email address with a call to Entrez. If you are going to perform a lot of queries consider setting `reutils.email` to your email address in your `.Rprofile` file.
- `reutils.show.headlines`: By default `efetch` objects containing text data show only the first 12 lines. This is quite handy if you have downloaded a fairly large genome in Genbank file format. This can be changed by setting the global option `reutils.show.headlines` to another numeric value or `NULL`.

- `reutils.verbose.queries`: If you perform many queries interactively you might want to get messages announcing the queries you run. You can do so by setting the option `reutils.verbose.queries` to `TRUE`.
- `reutils.test.remote`: Unit tests that require online access to NCBI services are disabled by default, as they cannot be guaranteed to be available/working under all circumstances. Set the option `codereutils.test.remote` to `TRUE` to run the full suite of tests.

Author(s)

Gerhard Schöfl <gerhard.schofl@gmail.com>

Examples

```
#
# combine esearch and efetch
#
# Download PubMed records that are indexed in MeSH for both 'Chlamydia' and
# 'genome' and were published in 2013.
query <- "Chlamydia[mesh] and genome[mesh] and 2013[pdat]"

# Upload the PMIDs for this search to the History server
pmids <- esearch(query, "pubmed", usehistory = TRUE)
pmids

## Not run:
# Fetch the records
articles <- efetch(pmids)

# Use XPath expressions with the #xmlValue() or #xmlAttr() methods to directly
# extract specific data from the XML records stored in the 'efetch' object.
titles <- articles$xmlValue("//ArticleTitle")
abstracts <- articles$xmlValue("//AbstractText")

#
# combine epost with esummary/efetch
#
# Download protein records corresponding to a list of GI numbers.
uid <- c("194680922", "50978626", "28558982", "9507199", "6678417")

# post the GI numbers to the Entrez history server
p <- epost(uid, "protein")

# retrieve docsums with esummary
docsum <- content(esummary(p, version = "1.0"), "parsed")
docsum

# download FASTAs as 'text' with efetch
prot <- efetch(p, retmode = "text", rettype = "fasta")
prot

# retrieve the content from the efetch object
fasta <- content(prot)
```

```
## End(Not run)
```

content

Extract the data content from an Entrez request

Description

There are five ways to access data returned by an Entrez request: as a character string (`as = "text"`), as a `textConnection` (`as = "textConnection"`), as an `XMLInternalDocument` (`as = "xml"`) or json object (`as = "json"`) (depending on the `retmode` with which the request was performed), or parsed into a native R object, e.g. a list or a `data.frame` (`as = "parsed"`).

Usage

```
content(x, ...)  
  
## S4 method for signature 'eutil'  
content(x, ...)  
  
## S4 method for signature 'ecitmatch'  
content(x, as = "text")  
  
## S4 method for signature 'efetch'  
content(x, as = NULL)  
  
## S4 method for signature 'egquery'  
content(x, as = NULL)  
  
## S4 method for signature 'einfo'  
content(x, as = NULL)  
  
## S4 method for signature 'elink'  
content(x, as = NULL)  
  
## S4 method for signature 'epost'  
content(x, as = NULL)  
  
## S4 method for signature 'esearch'  
content(x, as = NULL)  
  
## S4 method for signature 'espell'  
content(x, as = NULL)  
  
## S4 method for signature 'esummary'  
content(x, as = NULL)
```

Arguments

x	An eutil object.
...	Further arguments passed on to methods.
as	Type of output: "text", "xml", "json", "textConnection", or "parsed". content attempts to figure out the most appropriate output type, based on the retmode of the object.

Methods (by class)

- `eutil`: Access the data content from an `eutil` object.
- `ecitmatch`: Return PubMed IDs if `as = "parsed"`.
- `efetch`: Access the data content from an `efetch` request.
- `egquery`: Access the data content from an `egquery` request.
- `einfo`: Access the data content from an `einfo` request.
- `elink`: Access the data content from an `elink` request.
- `epost`: Access the data content from an `epost` request.
- `esearch`: Access the data content from an `esearch` request.
- `espell`: Access the data content from an `espell` request.
- `esummary`: Access the data content from an `esummary` request.

See Also

[einfo](#), [esearch](#), [esummary](#), [efetch](#), [elink](#), [epost](#), [egquery](#), [espell](#), [ecitmatch](#).

Examples

```
## Not run:
## einfo() defaults to retmode 'xml'
e <- einfo()

## automatically return data as an 'XMLInternalDocument'.
if (e$no_errors()) {
  content(e)

  ## return the XML data as character string.
  cat(content(e, "text"))

  ## return DbNames parsed into a character vector.
  content(e, "parsed")
}

## return data as a JSON object
e2 <- einfo(db = "gene", retmode = "json")
if (e2$no_errors()) {
  content(e2)
}
```

```
## return a textConnection to allow linewise reading of the data.
x <- efetch("CP000828", "nucore", rettype = "gbwithparts", retmode = "text")
con <- content(x, as = "textConnection")
readLines(con, 2)
close(con)

## End(Not run)
```

database

database

Description

Retrieve the target database name from an [eutil](#) object.

Usage

```
database(x, ...)
```

S4 method for signature 'eutil'

```
database(x, ...)
```

S4 method for signature 'entrez_linkset'

```
database(x, ...)
```

S4 method for signature 'entrez_uid'

```
database(x, ...)
```

Arguments

x An [eutil](#) object.

... Further arguments passed on to methods.

Value

A character string.

Methods (by class)

- `eutil`: Retrieve the target database name from an `eutil` object.
- `entrez_linkset`: Retrieve the target database name from an `entrez_linkset` object.
- `entrez_uid`: Retrieve the target database name from an `entrez_uid` object.

See Also

[einfo](#), [esearch](#), [esummary](#), [efetch](#), [elink](#), [epost](#), [egquery](#), [espell](#), [ecitmatch](#).

Examples

```
## Not run:
e <- esearch("Mus musculus", "taxonomy")
database(e)

## End(Not run)
```

ecitmatch

ecitmatch - searching PubMed with citation data

Description

ecitmatch serves as an API to the PubMed [batch citation matcher](#). It retrieves PubMed IDs (PMIDs) that correspond to a set of input citation strings.

Usage

```
ecitmatch(bdata, db = "pubmed", retmode = "xml")
```

Arguments

bdata	Citation strings. Each input citation must be represented by a citation string in the following format: <i>journal_title year volume first_page author_name your_key </i>
db	Database to search. The only supported value is 'pubmed'.
retmode	Retrieval mode. The only supported value is 'xml'.

Value

An [ecitmatch](#) object.

Examples

```
citstrings <- c("proc natl acad sci u s a|1991|88|3248|mann bj|Art1|",
               "science|1987|235|182|palmenber ac|Art2|")
x <- ecitmatch(citstrings)
x
if (x$no_errors()) {
  content(x, "parsed")
}
```

efetch	<i>efetch - downloading full records</i>
--------	--

Description

efetch performs calls to the NCBI EFetch utility to retrieve data records in the requested format for an NCBI Accession Number, one or more primary UIDs, or for a set of UIDs stored in the user's web environment.

Usage

```
efetch(uid, db = NULL, rettype = NULL, retmode = NULL, outfile = NULL,
       retstart = NULL, retmax = NULL, querykey = NULL, webenv = NULL,
       strand = NULL, seqstart = NULL, seqstop = NULL, complexity = NULL)
```

Arguments

uid	(Required) A list of UIDs provided either as a character vector, as an <code>esearch</code> object, or by reference to a Web Environment and a query key obtained directly from previous calls to <code>esearch</code> (if <code>usehistory = TRUE</code>), <code>epost</code> or <code>elink</code> . If UIDs are provided as a plain character vector, <code>db</code> must be specified explicitly, and all of the UIDs must be from the database specified by <code>db</code> .
db	(Required if <code>uid</code> is a character vector of UIDs) Database from which to retrieve records. See here for the supported databases.
rettype	A character string specifying the retrieval type, such as 'abstract' or 'medline' for PubMed, 'gp' or 'fasta' for Protein, or 'gb', or 'fasta' for Nucleotide. See here for the available values for each database.
retmode	A character string specifying the data mode of the records returned, such as 'text' or 'xml'. See here for the available values for each database.
outfile	A character string naming a file for writing the data to. Required if more than 500 UIDs are retrieved at once. In this case UIDs have to be provided by reference to a Web Environment and a query key obtained directly from previous calls to <code>esearch</code> (if <code>usehistory = TRUE</code>), <code>epost</code> or <code>elink</code> .
retstart	Numeric index of the first record to be retrieved.
retmax	Total number of records from the input set to be retrieved.
querykey	An integer specifying which of the UID lists attached to a user's Web Environment will be used as input to <code>efetch</code> . (Usually obtained directly from objects returned by a previous call to <code>esearch</code> , <code>epost</code> or <code>elink</code> .)
webenv	A character string specifying the Web Environment that contains the UID list. (Usually obtained directly from objects returned by a previous call to <code>esearch</code> , <code>epost</code> or <code>elink</code> .)
strand	Strand of DNA to retrieve. (1: plus strand, 2: minus strand)
seqstart	First sequence base to retrieve.
seqstop	Last sequence base to retrieve.

complexity Data content to return. (0: entire data structure, 1: bioseq, 2: minimal bioseq-set, 3: minimal nuc-prot, 4: minimal pub-set)

Details

See the official online documentation for NCBI's [EUtilities](#) for additional information.

See [here](#) for the default values for `rettype` and `retmode`, as well as a list of the available databases for the EFetch utility.

Value

An [efetch](#) object.

Note

If you are going to retrieve more than 500 UIDs at once, you will have to provide the UIDs by reference to a Web Environment and a query key obtained from previous calls to [esearch](#) (if `usehistory = TRUE`), [epost](#) or [elink](#) **and** you will have to specify an outfile to write the data to, rather than collecting the data into an R object.

See Also

[content](#), [getUrl](#), [getError](#), [database](#), [retmode](#), [rettype](#).

Examples

```
## Not run:
## From Protein, retrieve a raw GenPept record and write it to a file.
p <- efetch("195055", "protein", "gp")
p

write(content(p, "text"), file = "~/AAD15290.gp")

## Get accessions for a list of GenBank IDs (GIs)
acc <- efetch(c("1621261", "89318838", "68536103", "20807972", "730439"),
             "protein", rettype = "acc")
acc
acc <- strsplit(content(acc), "\n")[[1]]
acc

## Get GIs from a list of accession numbers
gi <- efetch(c("CAB02640.1", "EAS10332.1", "YP_250808.1", "NP_623143.1", "P41007.1"),
            "protein", "uilst")
gi

## we can conveniently extract the UIDs using the util method #xmlValue(xpath)
gi$xmlValue("/IdList/Id")

## or we can extract the contents of the efetch query using the fuction content()
## and use the XML package to retrieve the UIDs
doc <- content(gi)
```

```

XML::xpathSApply(doc, "/IdList/Id", XML::xmlValue)

## Get the scientific name for an organism starting with the NCBI taxon id.
tx <- efetch("527031", "taxonomy")
tx

## Convenience accessor for XML nodes of interest using XPath
## Extract the TaxIds of the Lineage
tx["//LineageEx/Taxon/TaxId"]

## Use an XPath expression to extract the scientific name.
tx$xmlValue("/TaxaSet/Taxon/ScientificName")

## Iteratively retrieve a large number of records
# First store approx. 8400 UIDs on the History server.
uid <- esearch(term = "hexokinase", db = 'protein', usehistory = TRUE)
# Fetch the records and write to file in batches of 500.
efetch(uid, rettype = "fasta", retmode = "text", outfile = "~/tmp/hexokinases.fna")

## End(Not run)

```

egquery

egquery - performing a global Entrez search

Description

egquery retrieves the number of records in all Entrez databases for a single text query.

Usage

```
egquery(term)
```

Arguments

term A valid Entrez text query.

Details

See the official online documentation for NCBI's [EUtilities](#) for additional information.

Value

An [egquery](#) object.

Examples

```

## Determine the number of records for mouse in Entrez.
e <- egquery("mouse[orgn]")
e

```

einfo

einfo - getting database statistics and search fields

Description

einfo queries the NCBI EInfo utility to retrieve the names of all valid Entrez databases, or, if db is provided, to retrieve statistics for a single database, including lists of indexing fields and available link names. Version 2.0 data is requested by default.

Usage

```
einfo(db = NULL, version = "2.0", retmode = "xml")
```

Arguments

db	A valid NCBI database name. If NULL, a list of all current NCBI databases is returned.
version	Specifies version 2.0 EInfo XML. Set to NULL for the older version.
retmode	'xml' (default) or 'json'.

Details

See the official online documentation for NCBI's [EUtilities](#) for additional information.

Value

An [einfo](#) object.

See Also

[content](#), [getUrl](#), [getError](#).

Examples

```
## Not run:
## Fetch a list of all current Entrez database names
einfo()

## Fetch statistics for an Entrez database and parse
## the data into a data.frame
x <- einfo("gene")
if (x$no_errors()) {
  content(x, "parsed")
}

## Fetch statistics for an Entrez database in JSON format
```

```

## and parse the data into a list
x <- einfo("pubmed", retmode = "json")
if (x$no_errors()) {
  content(x, "parsed")
}

## End(Not run)

```

elink

elink - finding related data through Entrez links

Description

elink generates a list of UIDs in a specified Entrez database that are linked to a set of input UIDs in either the same or another database. For instance, the ELink utility can find Entrez gene records linked to records in Entrez Protein.

Usage

```

elink(uid, dbFrom = NULL, dbTo = NULL, linkname = NULL,
      usehistory = FALSE, cmd = "neighbor", correspondence = FALSE,
      querykey = NULL, webenv = NULL, term = NULL, holding = NULL,
      datatype = NULL, reldate = NULL, mindate = NULL, maxdate = NULL)

```

Arguments

uid	(Required) A character vector of UIDs.
dbFrom	Initial database containing the UIDs in the input list.
dbTo	Destination database from which to retrieve linked UIDs. If not provided links will be sought in the database containing the input UIDs.
linkname	Name of the Entrez link to retrieve. Every link in Entrez is given a name of the form <i>dbFrom_dbTo_subset</i> .
usehistory	If TRUE search results are stored directly in the user's Web environment so that they can be used in subsequent calls to esummary or efetch .
cmd	ELink command mode (default: 'neighbor'). See Details.
correspondence	if TRUE correspondence between query UIDs and destination UIDs is preserved.
querykey	Query key.
webenv	Web Environment.
term	Search query to limit the output set of linked UIDs.
holding	Name of LinkOut provider.
datatype	Type of date to limit the search. One of 'mdat' (modification date), 'pdat' (publication date) or 'edat' (Entrez date).
reldate	umber of days back for which search items are returned.
mindate	Minimum date of search range. Format YYYY/MM/DD.
maxdate	Maximum date of search range. Format YYYY/MM/DD.

Details

See the official online documentation for NCBI's [EUtilities](#) for additional information.

If dbTo and dbFrom are set to the same database, ELink will return neighbors within that database.

Elink commands (cmd) specify the function that elink will perform. Available commands are:

- **"neighbor"** (Default) ELink returns a set of UIDs in dbTo linked to the input UIDs in dbFrom.
- **"neighbor_score"** ELink returns a set of UIDs within the same database as the input UIDs along with similarity scores.
- **"neighbor_history"** ELink posts the output UIDs to the Entrez History server and returns a query_key and WebEnv parameter. Alternatively this is achieved by setting usehistory=TRUE

Value

An [elink](#) object.

See Also

Combine calls to ELink with other EUtils: [esummary](#), [efetch](#).

Accessor methods: [content](#), [getUrl](#), [getError](#), [database](#), [uid](#), [linkset](#),

Examples

```
## Find one set of Gene IDs linked to nuccore GIs 34577062 and 24475906
e <- elink(c("927442695", "312836839"), dbFrom = "nuccore", dbTo = "gene")
e

## Not run:
## Find related articles to PMID 20210808
p <- elink("20210808", dbFrom = "pubmed", dbTo = "pubmed")
p

## Extract linked UIDs from the "pubmed" to "pubmed_reviews" link
linkset(p, "pubmed_pubmed_reviews")

## or
p["pubmed_pubmed_reviews"]

## retrieve the abstracts for the first five linked reviews
abstracts <- efetch(p["pubmed_pubmed_reviews"][1:5], rettype = "abstract")

## End(Not run)
```

epost

epost - uploading UIDs to Entrez

Description

epost uses the Entrez EPost utility to upload primary UIDs to the Entrez History server or append a list of UIDs to an existing set of UIDs attached to a Web Environment.

Usage

```
epost(uid, db = NULL, webenv = NULL)
```

Arguments

uid	(Required) List of UIDs provided as a character or as an esearch object.
db	(Required if uid is a character vector) Database containing the UIDs in the input list.
webenv	(Optional) Web Environment. If provided, this parameter specifies the Web Environment that will receive the UIDs sent by epost. epost will create a new query key associated with that Web Environment. The webenv value is usually returned by a previous call to esearch , epost or elink . If no webenv parameter is provided, the EPost utility will create a new Web Environment and post the UIDs to query key 1.

Details

epost returns an integer label called a query key and an encoded cookie string called a Web environment. epost objects can then be used instead of a UID list in subsequent calls to [esummary](#), [efetch](#), or [elink](#).

See the official online documentation for NCBI's [EUtilities](#) for additional information.

Value

An [epost](#) object.

Examples

```
## post a list of protein GIs to the Entrez History server
gi <- c("194680922", "50978626", "28558982", "9507199", "6678417")
p <- epost(gi, "protein")
p
```

 esearch

esearch - searching an Entrez database

Description

esearch performs searches using the the NCBI ESearch utility to retrieve primary UIDs matching a text query. These UIDs can be used in subsequent calls to [esummary](#), [efetch](#), or [elink](#).

Usage

```
esearch(term, db = "nucore", rettype = "uilst", retmode = "xml",
        retstart = 0, retmax = 100, usehistory = FALSE, webenv = NULL,
        querykey = NULL, sort = NULL, field = NULL, datatype = NULL,
        reldate = NULL, mindate = NULL, maxdate = NULL)
```

Arguments

term	A valid Entrez text query.
db	Database to search (default: nucore).
rettype	Retrieval type. (default: 'uilst', alternative: 'count')
retmode	Retrieval mode. (default: 'xml', alternative: 'json')
retstart	Numeric index of the first UID in the retrieved set to be shown in the XML output (default: 0).
retmax	Total number of UIDs to be retrieved (default: 100).
usehistory	If TRUE, search results are posted directly to the Entrez History Server so that they can be used in subsequent calls to esummary , efetch , or elink . Also, usehistory must be set to TRUE for esearch to interpret query key values included in term or to accept a webenv as input.
webenv	Web environment string returned by a previous call to esearch , epost or elink . When provided, esearch will append the results of the search to the pre-existing Web environment. Providing webenv also allows query keys to be used in term so that previous search sets can be combined or limited.
querykey	query key returned by a previous call to esearch , epost or elink . When provided, esearch will find the intersection of the set specified by querykey and the set retrieved by the query in term (i.e. joins the two with AND).
sort	Method used to sort UIDs in the ESearch output. The available values vary by database. Example values are 'relevance' and 'name' for Gene and 'first author' and 'pub date' for PubMed.
field	Optional. Search field used to limit the entire search term.
datatype	Optional. Type of date to limit the search. One of "mdat" (modification date), "pdat" (publication date) or "edat" (Entrez date)
reldate	Optional. Number of days back for which search items are returned.

mindate	Optional. Minimum date of search range. Format YYYY/MM/DD, YYYY/MM, or YYYY.
maxdate	Optional. Maximum date of search range. Format YYYY/MM/DD, YYYY/MM, or YYYY.

Details

See the official online documentation for NCBI's [EUtilities](#) for additional information on this EUtility.

Value

An [esearch](#) object.

See Also

Combine calls to ESearch with other EUtils: [esummary](#), [efetch](#), [elink](#).

Accessor methods: [content](#), [getUrl](#), [getError](#), [database](#), [uid](#), [webenv](#), [querykey](#).

Examples

```
## Search PubMed for articles with the term "Chlamydia psittaci" in the
## title that were published in 2013.
pmid <- esearch("Chlamydia psittaci[titl] and 2013[pdat]", "pubmed")
pmid

## Not run:
## Extract the query results either as an XML tree or parsed into
## a character vector
xml <- content(pmid, "xml")
uids <- uid(pmid)

## Alternatively post the UIDs to the History Server.
pmid <- esearch("Chlamydia psittaci[titl] and 2013[pdat]", "pubmed",
               usehistory = TRUE)
pmid

## Associate new search results with the existing search results.
pmid2 <- esearch("Chlamydia psittaci[titl] and 2012[pdat]", "pubmed",
               usehistory = TRUE, webenv = webenv(pmid))
pmid2

## Sort results by author
pmid3 <- esearch("Chlamydia psittaci[titl] and 2013[pdat]", "pubmed",
               sort = "first author")
pmid3

## End(Not run)
```

espell	<i>espell - retrieving spelling suggestions</i>
--------	---

Description

For a text query retrieve an XML containing the original query and spelling suggestions.

Usage

```
espell(term, db = "pubmed")
```

Arguments

term	An Entrez text query.
db	An Entrez database.

Value

An `espell` object.

Examples

```
e <- espell("Chlamidia")
e
```

esummary	<i>esummary - downloading Document Summaries</i>
----------	--

Description

esummary performs calls to the NCBI ESummary utility to retrieve document summaries (DocSums) for a list of primary UIDs or for a set of UIDs stored in the user's web environment (using the Entrez History server).

Usage

```
esummary(uid, db = NULL, retstart = 1, retmax = 10000, querykey = NULL,
  webenv = NULL, retmode = "xml", version = "2.0")
```

Arguments

uid	(Required) List of UIDs provided either as a character vector, as an <code>esearch</code> or <code>elink</code> object, or by reference to a Web Environment and a query key obtained directly from objects returned by previous calls to <code>esearch</code> , <code>epost</code> or <code>elink</code> . If UIDs are provided as a plain character vector, <code>db</code> must be specified explicitly, and all of the UIDs must be from the database specified by <code>db</code> .
db	(Required only when <code>id</code> is a character vector of UIDs) Database from which to retrieve DocSums.
retstart	Numeric index of the first DocSum to be retrieved (default: 1).
retmax	Total number of DocSums from the input set to be retrieved (maximum: 10,000).
querykey	An integer specifying which of the UID lists attached to a user's Web Environment will be used as input to <code>efetch</code> . (Usually obtained directly from objects returned by previous <code>esearch</code> , <code>epost</code> or <code>elink</code> calls.)
webenv	A character string specifying the Web Environment that contains the UID list. (Usually obtained directly from objects returned by previous <code>esearch</code> , <code>epost</code> or <code>elink</code> calls.)
retmode	Retrieval mode. (default: 'xml', alternative: 'json')
version	If "2.0" <code>esummary</code> will retrieve version 2.0 ESummary XML output.

Details

See the official online documentation for NCBI's [EUtilities](#) for additional information.

Value

An `esummary` object.

See Also

[content](#), [getUrl](#), [getError](#), [database](#).

Examples

```
## Retrieve the Document Summary information for a set of
## UIDs from the Nuccore datanase.
ds <- esummary(c("1060721643", "1060721620", "1060721618"), "nuccore")
ds

## Not run:
## parse the XML into a data frame
df <- content(ds, "parsed")
df

## use XPath expressions to extract nodes of interest
ds['//TaxId/text()']

## End(Not run)
```

`getError`*getError*

Description

Retrieve a http or XML parsing error from an `eutil` object.

Usage

```
getError(x, ...)  
  
## S4 method for signature 'eutil'  
getError(x, ...)
```

Arguments

<code>x</code>	An <code>eutil</code> object.
<code>...</code>	Further arguments passed on to methods.

Value

An `eutil_error` object.

Methods (by class)

- `eutil`: a http or XML parsing error from an `eutil` object.

See Also

[einfo](#), [esearch](#), [esummary](#), [efetch](#), [elink](#), [epost](#), [egquery](#), [espell](#), [ecitmatch](#).

Examples

```
## Not run:  
e <- efetch("Nonsensical_accession_nr", "protein", rettype = "fasta")  
getError(e)  
  
## End(Not run)
```

getUrl	<i>getUrl</i>
--------	---------------

Description

Retrieve the URL used to perform an Entrez E-Utilities query.

Usage

```
getUrl(x, ...)  
  
## S4 method for signature 'eutil'  
getUrl(x, ...)
```

Arguments

x	An eutil object.
...	Further arguments passed on to methods.

Value

A character string.

Methods (by class)

- `eutil`: retrieve the URL used to perform an Entrez E-Utilities query from an `eutil` object.

See Also

[einfo](#), [esearch](#), [esummary](#), [efetch](#), [elink](#), [epost](#), [egquery](#), [espell](#), [ecitmatch](#).

Examples

```
## Not run:  
e <- efetch("AV333213.1", "protein", rettype = "fasta")  
getUrl(e)  
  
## End(Not run)
```

linkset	<i>linkset</i>
---------	----------------

Description

Retrieve a linkset from an [elink](#) object.

Usage

```
linkset(x, linkname = NULL, ...)

## S4 method for signature 'entrez_linkset'
linkset(x, linkname = NULL, ...)

## S4 method for signature 'elink'
linkset(x, linkname = NULL, ...)
```

Arguments

x	An elink object.
linkname	(optional) Name of the Entrez link to retrieve. Every link in Entrez is given a name of the form <i>dbFrom_dbTo_subset</i> . If NULL, all available links are retrieved from the object.
...	Further arguments passed on to methods.

Value

A list.

Methods (by class)

- `entrez_linkset`: Retrieve a linkset from an [elink](#) object.
- `elink`: Retrieve a linkset from an [elink](#) object.

Examples

```
## Not run:
## Find related articles to PMID 20210808 and xtract linked UIDs from the
## "pubmed" to "pubmed_reviews" link
x <- elink("20210808", dbFrom = "pubmed", dbTo = "pubmed", cmd = "neighbor_score")
linkset(x, "pubmed_pubmed_reviews")

## End(Not run)
```

querykey	<i>querykey</i>
----------	-----------------

Description

An integer query key returned by an ESearch, EPost or ELink call if the History server was used. Otherwise NA.

Usage

```
querykey(x, ...)  
  
## S4 method for signature 'epost'  
querykey(x, ...)  
  
## S4 method for signature 'entrez_uid'  
querykey(x, ...)  
  
## S4 method for signature 'esearch'  
querykey(x, ...)
```

Arguments

x	An eutil object.
...	Further arguments passed on to methods.

Value

An integer or NA.

Methods (by class)

- epost: Retrieve the querykey from an epost object.
- entrez_uid: Retrieve the querykey from an entrez_uid object.
- esearch: Retrieve the querykey from an esearch object.

See Also

[einfo](#), [esearch](#), [esummary](#), [efetch](#), [elink](#), [epost](#), [egquery](#), [espell](#), [ecitmatch](#).

Examples

```
## Not run:  
e <- esearch("Mus musculus", "taxonomy", usehistory = TRUE)  
querykey(e)  
  
## End(Not run)
```

retmode	<i>retmode</i>
---------	----------------

Description

Get the “retrieval mode” of an [eutil](#) object. It is usually one of xml, json, text, or asn.1. It is set to NULL if “retrieval mode” is not supported by an E-Utility.

Usage

```
retmode(x, ...)  
  
## S4 method for signature 'eutil'  
retmode(x, ...)
```

Arguments

x	An eutil object.
...	Further arguments passed on to methods.

Value

A character string or NULL.

Methods (by class)

- [eutil](#): Access the “retrieval mode” of an [eutil](#) object.

See Also

[einfo](#), [esearch](#), [esummary](#), [efetch](#), [elink](#), [epost](#), [egquery](#), [espell](#), [ecitmatch](#).

Examples

```
## Not run:  
e <- efetch("10090", "taxonomy")  
retmode(e)  
  
## End(Not run)
```

rettype	<i>rettype</i>
---------	----------------

Description

Get the “retrieval type” of an [eutil](#) object. See [here](#) for the available retrieval types for different NCBI databases.

Usage

```
rettype(x, ...)  
  
## S4 method for signature 'eutil'  
rettype(x, ...)
```

Arguments

x	An eutil object.
...	Further arguments passed on to methods.

Value

A character string.

Methods (by class)

- [eutil](#): Access the “retrieval type” of an [eutil](#) object.

See Also

[einfo](#), [esearch](#), [esummary](#), [efetch](#), [elink](#), [epost](#), [egquery](#), [espell](#), [ecitmatch](#).

Examples

```
## Not run:  
e <- esearch("Mus musculus", "taxonomy")  
rettype(e)  
  
## End(Not run)
```

`uid`*uid*

Description

Retrieve the list of UIDs returned by a call to ESearch or ELink.

Usage

```
uid(x, ...)  
  
## S4 method for signature 'entrez_linkset'  
uid(x, ...)  
  
## S4 method for signature 'elink'  
uid(x, ...)  
  
## S4 method for signature 'entrez_uid'  
uid(x, ...)  
  
## S4 method for signature 'esearch'  
uid(x, ...)
```

Arguments

<code>x</code>	An eutil object.
<code>...</code>	Further arguments passed on to methods.

Value

A character vector.

Methods (by class)

- `entrez_linkset`: Retrieve the list of UIDs from an `entrez_linkset` object.
- `elink`: Retrieve the list of UIDs from an `elink` object.
- `entrez_uid`: Retrieve the list of UIDs from an `entrez_uid` object.
- `esearch`: Retrieve the list of UIDs from an `esearch` object.

See Also

[esearch](#), [elink](#).

Examples

```
## Not run:
e <- esearch("Mus musculus", "taxonomy")
uid(e)

## End(Not run)
```

webenv

webenv

Description

Retrieve the Web environment string returned from an ESearch, EPost or ELink call. NA if the History server was not used.

Usage

```
webenv(x, ...)
```

S4 method for signature 'epost'

```
webenv(x, ...)
```

S4 method for signature 'entrez_uid'

```
webenv(x, ...)
```

S4 method for signature 'esearch'

```
webenv(x, ...)
```

Arguments

x An [eutil](#) object.

... Further arguments passed on to methods.

Value

A character string or NA.

Methods (by class)

- epost: Retrieve the webenv string from an epost object.
- entrez_uid: Retrieve the webenv string from an entrez_uid object.
- esearch: Retrieve the webenv string from an esearch object.

See Also

[einfo](#), [esearch](#), [esummary](#), [efetch](#), [elink](#), [epost](#), [egquery](#), [espell](#), [ecitmatch](#).

Examples

```
## Not run:
e <- esearch("Mus musculus", "taxonomy", usehistory = TRUE)
webenv(e)

## End(Not run)
```

[,efetch,character,missing-method
EFetch accessors

Description

Extract XML nodes from an [efetch](#) object.

Usage

```
## S4 method for signature 'efetch,character,missing'
x[i, j]

## S4 method for signature 'efetch,character'
x[[i]]
```

Arguments

x	An efetch object containing XML data.
i	An XPath expression specifying the XML nodes to extract.
j	Ignored.

Value

An XML node set.

Examples

```
## Not run:
p <- efetch("195055", "protein", "gp", "xml")
p['//GBFeature[GBFeature_key="mat_peptide"]//GBQualifier_value']

## End(Not run)
```

[,einfo,ANY,missing-method
EInfo accessors

Description

Extract parts of a parsed [einfo](#) object.

Usage

```
## S4 method for signature 'einfo,ANY,missing'  
x[i, j]
```

```
## S4 method for signature 'einfo,ANY'  
x[[i]]
```

Arguments

x	An einfo object.
i	Numeric or character indices specifying the elements to extract.
j	Ignored.

Value

A list.

See Also

[Extract](#)

Examples

```
## Not run:  
e <- einfo("pubmed")  
e[1:5]  
e["Description"]  
e[["Links"]]  
  
e2 <- einfo("pubmed", retmode = 'json')  
e2[["header"]]  
e2[["einforesult"]][["dbinfo"]][["description"]]  
  
## End(Not run)
```

[,elink,ANY,missing-method

ELink Accessors

Description

Extract UIDs from an [elink](#) object.

Usage

```
## S4 method for signature 'elink,ANY,missing'  
x[i, j]
```

```
## S4 method for signature 'elink,character,ANY'  
x[i]
```

Arguments

x	An elink object.
i	Numeric or character indices.
j	Ignored.

Value

A [entrez_linkset](#) object.

Examples

```
## Not run:  
e <- elink(c("34577062", "24475906"), dbFrom = "nucore")  
e[1]  
  
## End(Not run)
```

[,esearch,numeric,missing-method

ESearch Accessors

Description

Extract UIDs from an [esearch](#) object.

Usage

```
## S4 method for signature 'esearch,numeric,missing'  
x[i, j]
```

Arguments

x	An esearch object.
i	Numeric indices.
j	Ignored.

Value

A [entrez_uid](#) object.

Examples

```
## Not run:
e <- esearch("Mus musculus", "protein", retmax = 20)
e[1:5]
## pass the subset directly on to esummary or efetch
content(esummary(e[1:5]), "parsed")

## End(Not run)
```

[,esummary,character,ANY-method
ESummary accessors

Description

Extract XML nodes from an [esummary](#) object.

Usage

```
## S4 method for signature 'esummary,character,ANY'
x[i]

## S4 method for signature 'esummary,character'
x[[i]]
```

Arguments

x	An esummary object.
i	An XPath expression.

Value

An XML node set.

Examples

```
## Not run:  
ds <- esummary("470338", "protein")  
ds["//Slen/node()"]  
  
as.numeric(XML::xmlValue(ds[["//Slen"]]))  
  
## End(Not run)
```

Index

- [,efetch,character,missing-method, 28
- [,einfo,ANY,missing-method, 29
- [,elink,ANY,missing-method, 30
- [,elink,character,ANY-method
 - ([,elink,ANY,missing-method), 30
- [,esearch,numeric,missing-method, 30
- [,esummary,character,ANY-method, 31
- [[,efetch,character-method
 - ([,efetch,character,missing-method), 28
- [[,einfo,ANY-method
 - ([,einfo,ANY,missing-method), 29
- [[,esummary,character-method
 - ([,esummary,character,ANY-method), 31
- content, 3, 5, 10, 12, 14, 17, 19
- content,ecitmatch-method (content), 5
- content,efetch-method (content), 5
- content,egquery-method (content), 5
- content,einfo-method (content), 5
- content,elink-method (content), 5
- content,epost-method (content), 5
- content,esearch-method (content), 5
- content,espell-method (content), 5
- content,esummary-method (content), 5
- content,eutil-method (content), 5
- database, 7, 10, 14, 17, 19
- database,entrez_linkset-method (database), 7
- database,entrez_uid-method (database), 7
- database,eutil-method (database), 7
- ecitmatch, 3, 6–8, 8, 20, 21, 23–25, 27
- efetch, 3, 6, 7, 9, 10, 13–17, 20, 21, 23–25, 27, 28
- egquery, 3, 6, 7, 11, 11, 20, 21, 23–25, 27
- einfo, 3, 6, 7, 12, 12, 20, 21, 23–25, 27, 29
- elink, 3, 6, 7, 9, 10, 13, 14–17, 19–27, 30
- entrez_linkset, 30
- entrez_uid, 31
- epost, 3, 6, 7, 9, 10, 15, 15, 16, 19–21, 23–25, 27
- esearch, 3, 6, 7, 9, 10, 15, 16, 16, 17, 19–21, 23–27, 30, 31
- espell, 3, 6, 7, 18, 18, 20, 21, 23–25, 27
- esummary, 3, 6, 7, 13–17, 18, 19–21, 23–25, 27, 31
- eutil, 3, 6, 7, 20, 21, 23–27
- eutil_error, 20
- Extract, 29
- getError, 10, 12, 14, 17, 19, 20
- getError,eutil-method (getError), 20
- getUrl, 10, 12, 14, 17, 19, 21
- getUrl,eutil-method (getUrl), 21
- linkset, 14, 22
- linkset,elink-method (linkset), 22
- linkset,entrez_linkset-method (linkset), 22
- options, 3
- querykey, 17, 23
- querykey,entrez_uid-method (querykey), 23
- querykey,epost-method (querykey), 23
- querykey,esearch-method (querykey), 23
- retmode, 10, 24
- retmode,eutil-method (retmode), 24
- rettype, 10, 25
- rettype,eutil-method (rettype), 25
- reutils (reutils-package), 2
- reutils-package, 2
- textConnection, 5

uid, [14](#), [17](#), [26](#)

uid,elink-method (uid), [26](#)

uid,entrez_linkset-method (uid), [26](#)

uid,entrez_uid-method (uid), [26](#)

uid,esearch-method (uid), [26](#)

webenv, [17](#), [27](#)

webenv,entrez_uid-method (webenv), [27](#)

webenv,epost-method (webenv), [27](#)

webenv,esearch-method (webenv), [27](#)

XMLInternalDocument, [5](#)