

# Package ‘pubmedtk’

May 9, 2026

**Title** 'Pubmed' Toolkit

**Version** 1.0.4

**Description** Provides various functions for retrieving and interpreting information from 'Pubmed' via the API, <<https://www.ncbi.nlm.nih.gov/home/develop/api/>>.

**License** AGPL (>= 3)

**URL** <https://github.com/bgcarlisle/pubmedtk>

**BugReports** <https://github.com/bgcarlisle/pubmedtk/issues>

**Imports** magrittr, dplyr, xml2, jsonlite, httr, assertthat, stringr, rlang, utils

**Encoding** UTF-8

**RoxygenNote** 7.3.1

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Benjamin Gregory Carlisle [aut, cre] (ORCID: <<https://orcid.org/0000-0001-8975-0649>>)

**Maintainer** Benjamin Gregory Carlisle <murph@bgcarlisle.com>

**Repository** CRAN

**Date/Publication** 2024-03-05 21:50:02 UTC

## Contents

get_metadata_from_one_pmids . . . . .	2
get_metadata_from_pmids . . . . .	3
get_pmids_from_one_search . . . . .	5
get_pmids_from_searches . . . . .	6
intersection_check . . . . .	7

<b>Index</b>	<b>9</b>
--------------	----------

get\_metadata\_from\_one\_pmid

*Downloads metadata from Pubmed API for a single provided PMID and exports*

---

### **Description**

Downloads metadata from Pubmed API for a single provided PMID and exports

### **Usage**

```
get_metadata_from_one_pmid(pmid, api_key)
```

### **Arguments**

pmid	A single PMID, e.g. "29559429"
api_key	A valid Pubmed API key

### **Value**

A named list with 7 elements:

``$pubmed_dl_success``, which is TRUE in the case that a corresponding Pubmed record was found and metadata downloaded and FALSE otherwise.

``$doi``, a character string containing the DOI for the publication with the PMID in question.

``$languages``, a list of languages corresponding to the publication with the PMID in question.

``$pubtypes``, a list of publication types corresponding to the publication with the PMID in question.

``$pubdate``, the listed publication date

``$epubdate``, the listed e-publication date

``$authors``, a list of authors of the publication with the PMID in question.

``$abstract``, a character string containing the abstract for the publication with the PMID in question.

**Examples**

```
## Not run:
## Read in API key
ak <- readLines("api_key.txt")

## Download Pubmed metadata
mdata <- get_metadata_from_one_pmid("29559429", ak)

## Extract first author
mdata$authors[1]

## End(Not run)
```

---

```
get_metadata_from_pmids
```

*Downloads metadata from Pubmed API for a column of PMID's in a data frame*

---

**Description**

Downloads metadata from Pubmed API for a column of PMID's in a data frame

**Usage**

```
get_metadata_from_pmids(df, column, api_key, quiet = FALSE)
```

**Arguments**

df	A dataframe containing a column of PMID's
column	The name of the column containing PMID's
api_key	A valid Pubmed API key
quiet	A boolean TRUE or FALSE. If TRUE, no progress messages will be printed during download. FALSE by default, messages printed for every version downloaded showing progress.

**Value**

A data frame containing the original columns as well as seven additional columns:

The ``pubmed_dl_success`` column is TRUE in the case that metadata were successfully downloaded from Pubmed; FALSE in the case that an error occurred during downloading (e.g. due to a number that is well-formed but does not correspond to a true PMID); NA in the case that the supplied PMID is not well-formed (e.g. NA or non-numeric).

The ``doi`` column returns a DOI that corresponds to the PMID

supplied if one is found, NA otherwise.

The ``languages`` column contains a JSON-encoded list of languages for the article in question.

The ``pubtypes`` column contains a JSON-encoded list of publication types for the article in question.

The ``pubdate`` column contains a character string with the publication date

The ``epubdate`` column contains a character string with the e-publication date

The ``authors`` column contains a JSON-encoded list of authors for the article in question.

The ``abstract`` column contains a character string with the abstract for the article in question.

## Examples

```
## Not run:
## Read in API key
ak <- readLines("api_key.txt")

## Example publications and their corresponding PMID's (some valid
## and some not)
pubs <- tibble::tribble(
  ~pmid,
  "29559429",
  "28837722",
  NA,
  "borp",
  "98472657638729"
)

## Download Pubmed metadata
pm_meta <- get_metadata_from_pmids(pubs, "pmid", ak)

## Extract DOI's for those that were successfully downloaded
pm_meta %>%
  dplyr::filter(pubmed_dl_success)
  dplyr::select(pmid, doi)

## A tibble: 2 × 2
##   pmid    doi
##   <chr>  <chr>
## 1 29559429 10.1136/bmj.k959
## 2 28837722 10.1001/jama.2017.11502
```

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

---

```
get_pmids_from_one_search
```

*Returns a list of PMID's for a provided Pubmed search*

---

## Description

Returns a list of PMID's for a provided Pubmed search

## Usage

```
get_pmids_from_one_search(query, api_key)
```

## Arguments

query	A Pubmed search query
api_key	A valid Pubmed API key

## Value

A named list with 3 elements:

``$pubmed_search_success``, which is TRUE in the case that the provided query was searched successfully on Pubmed and FALSE otherwise.

``$n_results``, the number of results for the search as reported by Pubmed

``$pmids``, a list of PMID's corresponding to the Pubmed search results for the query provided

## Examples

```
## Not run:
## Read in API key
ak <- readLines("api_key.txt")

## Download PMID's for search query
results <- get_pmids_from_one_search("Carlisle B[Author]", ak)

## Extract first result
results$pmids[1]

## End(Not run)
```

---

`get_pmids_from_searches`

*Downloads PMID results for a column of Pubmed search queries in a data frame*

---

### Description

Downloads PMID results for a column of Pubmed search queries in a data frame

### Usage

```
get_pmids_from_searches(df, column, api_key, quiet = FALSE)
```

### Arguments

<code>df</code>	A dataframe containing a column of Pubmed search queries. This data frame cannot have columns with the following names: <code>pubmed_search_success</code> , <code>pmids</code> .
<code>column</code>	The name of the column containing Pubmed search queries
<code>api_key</code>	A valid Pubmed API key
<code>quiet</code>	A boolean TRUE or FALSE. If TRUE, no progress messages will be printed during download. FALSE by default, messages printed for every version downloaded showing progress.

### Value

A data frame containing the original columns as well as three additional columns:

The ``pubmed_search_success`` column is TRUE in the case that the search results were successfully obtained from Pubmed; FALSE in the case that an error occurred in search (e.g. due to a search query that is not well-formed).

The ``n_results`` column contains the number of research results for the query provided.

The ``pmids`` column returns a JSON-encoded list of PMID's for the search query provided.

### Examples

```
## Not run:
## Read in API key
ak <- readLines("api_key.txt")

## Example Pubmed searches, some valid, some not, some with more
## than 10k results
searches <- tribble(
```

```

    ~terms,
    "Carlisle B[Author]",
    "NCT00267865",
    "(Clinical Trial[Publication Type])",
    ""
  )

## Download search results
results <- get_pmids_from_searches(searches, "terms", ak)

## End(Not run)

```

---

intersection_check	<i>Checks a column of PMID's for whether or not they would appear in a Pubmed search result.</i>
--------------------	--

---

## Description

Checks a column of PMID's for whether or not they would appear in a Pubmed search result.

## Usage

```

intersection_check(
  df,
  column,
  query,
  api_key,
  batch_size = 1000,
  quiet = FALSE
)

```

## Arguments

df	A dataframe containing a column of PMID's
column	The name of the column containing PMID's
query	A character string containing a valid Pubmed search query
api_key	A valid Pubmed API key
batch_size	An integer greater than 0 and less than 10000
quiet	A boolean TRUE or FALSE. If TRUE, no progress messages will be printed during download. FALSE by default.

## Value

A data frame containing the original columns, as well as two additional ones: pm\_checked and found\_in\_pm\_query.

The new ``pm_checked`` column is TRUE if Pubmed was successfully queried and NA if Pubmed was not checked for that PMID (this may occur in cases where the PMID to be checked is not well-formed).

The new ``found_in_pm_query`` column is TRUE if the PMID in question would appear in a search of Pubmed defined by the query provided; FALSE if it would not appear in such a search and NA if the PMID in question was not checked (this may occur in cases where the PMID is not well-formed).

### Examples

```
## Not run:
## Read in API key
ak <- readLines("api_key.txt")

## Example publications and their corresponding PMID's (some valid
## and some not)
pubs <- tibble::tribble(
  ~pmid,
  "29559429",
  "28837722",
  "28961465",
  "32278621",
  "one hundred of them",
  "28837722",
  "28961465"
)

## Check which ones were authored by Carlisle:
intersection_check(pubs, "pmid", "Carlisle[Author]", ak)

## End(Not run)
```

# Index

`get_metadata_from_one_pmid`, [2](#)

`get_metadata_from_pmids`, [3](#)

`get_pmids_from_one_search`, [5](#)

`get_pmids_from_searches`, [6](#)

`intersection_check`, [7](#)