

# Package ‘LitFetchR’

May 7, 2026

**Type** Package

**Title** Automatically Fetching References Metadata from Literature Databases

**Version** 0.2.2

**Maintainer** Thomas Dumond <thomas.dumond@adelaide.edu.au>

**Description** Provides functions to automatically retrieve and deduplicate reference metadata based on saved search strings. Access to Web of Science and Scopus requires personal API keys, while PubMed can be queried without one. The optional deduplication functionality requires the package 'ASySD' available from <<https://github.com/camaradesuk/ASySD>>.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Imports** dplyr, httr, jsonlite, openxlsx, purrr, readxl, xml2

**Suggests** ASySD, cronR, knitr, rmarkdown, taskscheduleR, testthat (>= 3.0.0)

**RoxygenNote** 7.3.3

**Config/testthat/edition** 3

**URL** <https://github.com/thomasdumond/LitFetchR>,  
<https://thomasdumond.github.io/LitFetchR/>

**BugReports** <https://github.com/thomasdumond/LitFetchR/issues>

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Thomas Dumond [cre, aut, cph] (ORCID: <<https://orcid.org/0000-0002-9427-8649>>),  
Charles Caraguel [ctb] (ORCID: <<https://orcid.org/0000-0003-0019-4813>>),  
Torben Nielsen [ctb] (ORCID: <<https://orcid.org/0000-0002-8941-7376>>)

**Repository** CRAN

**Date/Publication** 2026-04-14 07:10:02 UTC

## Contents

auto_LitFetchR_setup . . . . .	2
create_save_search . . . . .	3
dedup_refs . . . . .	4
manual_fetch . . . . .	5
remove_scheduled_task . . . . .	6
save_api_keys . . . . .	7
<b>Index</b>	<b>8</b>

---

auto\_LitFetchR\_setup *Automating the retrieval of references based on a saved search string(s).*

---

## Description

Creates a read-only Rscript and a task to run the code automatically at a specified frequency and time, to retrieve references corresponding to the saved search string(s) on up to three platforms (e.g. [Web of Science](#), [Scopus](#) and [PubMed](#)).

## Usage

```
auto_LitFetchR_setup(
  task_id = "task_id",
  when = "DAILY",
  time = "08:00",
  wos = FALSE,
  scp = FALSE,
  pmd = FALSE,
  directory,
  dedup = FALSE,
  open_file = FALSE,
  dry_run = FALSE
)
```

## Arguments

task_id	Name of the automated reference retrieval task (e.g. one keyword describing your review).
when	Frequency of the automated reference retrieval task (DAILY, WEEKLY or MONTHLY).
time	Time of the automated reference retrieval task (must be HH:MM 24-hour clock format).
wos	Runs the search on Web of Science (TRUE or FALSE).
scp	Runs the search on Scopus (TRUE or FALSE).
pmd	Runs the search on PubMed (TRUE or FALSE).

directory	Choose the directory in which the search string is saved (Project's directory). That is also where the references metadata will be saved.
dedup	Deduplicates the retrieved references (TRUE or FALSE).
open_file	Automatically opens the CSV file after reference retrieval.
dry_run	Simulation run option.

### Value

NULL (invisibly). Called for its side effects: writes an R script and schedules a task (Windows Task Scheduler or cron) to run the script automatically.

### Examples

```
# This is a "dry run" example.
# No task will actually be scheduled,
# it only shows how the function should react.
auto_LitFetchR_setup(task_id = "fish_vibrio",
                      when = "WEEKLY",
                      time = "14:00",
                      wos = TRUE,
                      scp = TRUE,
                      pmd = TRUE,
                      directory,
                      dedup = FALSE,
                      open_file = FALSE,
                      dry_run = TRUE
                      )
```

---

create\_save\_search      *Creates and saves search string(s).*

---

### Description

An interactive function that ask the user to enter a search string and provide the number of results from 3 platforms: Web of Science, Scopus and PubMed. You can then save one or more search strings to retrieve the references later.

### Usage

```
create_save_search(
  wos = FALSE,
  scp = FALSE,
  pmd = FALSE,
  directory,
  dry_run = FALSE
)
```

**Arguments**

wos	Runs the search on Web of Science (TRUE or FALSE).
scp	Runs the search on Scopus (TRUE or FALSE).
pmd	Runs the search on PubMed (TRUE or FALSE).
directory	Choose the directory in which the search string and the search history will be saved.
dry_run	Simulation run option.

**Value**

NULL (invisibly). Called for its side effects: interactive querying and writing search history files.

**Examples**

```
# This is a "dry run" example.  
# No search will be created and no database will be accessed.  
# It only shows how the function should react.  
create_save_search(wos = TRUE,  
                  scp = TRUE,  
                  pmd = TRUE,  
                  directory,  
                  dry_run = TRUE)
```

---

dedup\_refs

*Deduplicates the references from up to three dataframes.*

---

**Description**

Deduplicates the references from up to three dataframes.

**Usage**

```
dedup_refs(  
  df1 = NULL,  
  df2 = NULL,  
  df3 = NULL,  
  directory,  
  open_file = FALSE,  
  dry_run = FALSE  
)
```

### Arguments

df1	Dataframe 1 (can be NULL)
df2	Dataframe 2 (can be NULL)
df3	Dataframe 3 (can be NULL)
directory	Choose the directory in which the references deduplication history will be saved.
open_file	Automatically opens the CSV file after reference retrieval.
dry_run	Simulation run option.

### Value

NULL (invisibly). Called for its side effects: writes a CSV of deduplicated citations and an Excel workbook recording the deduplication history.

### Examples

```
# This is a "dry run" example.
# No deduplication will happen.
# It only shows how the function should react.
dedup_refs(df1 = df_vibrio_wos,
           df2 = df_vibrio_scp,
           df3 = df_vibrio_pmd,
           directory = tempdir(),
           open_file = FALSE,
           dry_run = TRUE
           )
```

---

manual\_fetch

*Manual literature retrieval.*

---

### Description

Retrieves references corresponding to the saved search string(s) on up to three platforms (e.g. [Web of Science](#), [Scopus](#) and [PubMed](#)).

### Usage

```
manual_fetch(
  wos = FALSE,
  scp = FALSE,
  pmd = FALSE,
  directory,
  dedup = FALSE,
  open_file = FALSE,
  dry_run = FALSE
)
```

**Arguments**

wos	Runs the search on Web of Science (TRUE or FALSE).
scp	Runs the search on Scopus (TRUE or FALSE).
pmd	Runs the search on PubMed (TRUE or FALSE).
directory	Choose the directory in which the search string is saved (Project's directory). That is also where the references metadata will be saved.
dedup	Deduplicates the retrieved references (TRUE or FALSE).
open_file	Automatically opens the CSV file after reference retrieval.
dry_run	Simulation run option.

**Value**

NULL (invisibly). Called for its side effects: Create a CSV file with the references metadata, a history file of the references retrieved and a history file of the deduplication (if the option is selected).

**Examples**

```
# This is a "dry run" example.
# No references will actually be scheduled, it only shows how the function should react.
manual_fetch(wos = TRUE,
             scp = TRUE,
             pmd = TRUE,
             directory,
             dedup = TRUE,
             open_file = FALSE,
             dry_run = TRUE
            )
```

---

remove\_scheduled\_task *Removes a scheduled task.*

---

**Description**

Removes a scheduled task using the "task\_id" from Task Scheduler (Windows) or Cron (Mac/Linux).

**Usage**

```
remove_scheduled_task(task_id, dry_run = FALSE)
```

**Arguments**

task_id	Name/ID of the scheduled task (Windows Task Scheduler or Cron).
dry_run	Simulation run option.

**Value**

NULL (invisibly). Called for its side effects: removes a scheduled task saved using the function 'auto\_LitFetchR\_setup'.

**Examples**

```
# This is a "dry run" example.  
# No task will actually be removed, it only shows how the function should react.  
remove_scheduled_task("fish_vibrio",  
                      dry_run = TRUE  
                      )
```

---

save_api_keys	<i>Saves Web of Science and/or Scopus API keys in .Renviron.</i>
---------------	--

---

**Description**

You can set wos\_api\_key, scp\_api\_key, or both at the same time. Remember to restart the R session after saving your API keys.

**Usage**

```
save_api_keys(wos_api_key = NULL, scp_api_key = NULL, dry_run = FALSE)
```

**Arguments**

wos_api_key	The API key value for Web of Science (use quotation marks).
scp_api_key	The API key value for Scopus (use quotation marks).
dry_run	Simulation run option.

**Value**

Logical. TRUE if at least one value was written, FALSE if left unchanged.

**Examples**

```
save_api_keys(wos_api_key = "abcd01234",  
             scp_api_key = "efgh5678",  
             dry_run = TRUE  
             )
```

# Index

`auto_LitFetchR_setup`, [2](#)

`create_save_search`, [3](#)

`dedup_refs`, [4](#)

`manual_fetch`, [5](#)

`remove_scheduled_task`, [6](#)

`save_api_keys`, [7](#)