

# Package: forgescraper (via r-universe)

June 10, 2026

**Version** 0.1-0

**Date** 2026-06-10

**Title** R-Forge Forum Scraper (Saver)

**Description** Small package for scraping the content of specific R-Forge forums and converting it into a static Quarto website.

**Depends** R (>= 4.4.0), xml2 (>= 1.5), httr2 (>= 1.2), yaml (>= 2.3), tidy

**License** GPL-2 | GPL-3

**URL** <https://codeberg.org/retostauffer/forgescraper>

**BugReports** <https://codeberg.org/zeileis/forgescraper/issues>

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.3

**Config/pak/sysreqs** libicu-dev libxml2-dev libssl-dev

**Repository** <https://retostauffer.r-universe.dev>

**Date/Publication** 2026-06-10 08:33:35 UTC

**RemoteUrl** <http://codeberg.org/retostauffer/forgescraper>

**RemoteRef** HEAD

**RemoteSha** 4ad645a8ebca0cec9736a55006f1bdfe2563891b

## Contents

download_attachment . . . . .	2
fs_get_all_threads . . . . .	2
fs_get_config . . . . .	3
fs_get_forum_details . . . . .	4
fs_get_forums . . . . .	5
fs_get_messages . . . . .	5
fs_write_quarto . . . . .	6
get_attachment_info . . . . .	7

<b>Index</b>	<b>8</b>
--------------	----------

download\_attachment     *Downloading Attachment*

---

**Description**

Helper function to download and store attachments.

Helper function to download and store attachments.

**Usage**

```
download_attachment(x, dir, verbose = FALSE)
```

```
download_attachment(x, dir, verbose = FALSE)
```

**Arguments**

x	named list with attachment name, url, and attachment_id (int).
dir	character, name of the 'thread output directory'.
verbose	logical, if set TRUE some output is shown.

**Value**

Invisibly returns the result of the request (typically unused).

Invisibly returns the result of the request (typically unused).

**Author(s)**

Reto

---

fs\_get\_all\_threads     *Downloading Forum Topics*

---

**Description**

Downloading Forum Topics

**Usage**

```
fs_get_all_threads(config, forums, verbose = TRUE)
```

**Arguments**

config	an object of class "fs_config" as returned by <code>fs_get_config()</code> .
forums	object of class "fs_forums" as returned by <code>fs_get_forums()</code> .
verbose	logical, defaults to TRUE.

**Value**

Returns a list of tibble data frame with all threads for the forums listed in forums.

This information is also stored in a file `project_{PROJECT}/threads_{forum_id}.rds`. If that file already exists, the content of the file (tibble data frame) is loaded from disc. Else, the website (forum) is scraped, the result stored to the RDS file mentioned above and re-used if the function is called next time.

**Author(s)**

Reto

---

<code>fs_get_config</code>	<i>R-Forge Forum Scraper Config</i>
----------------------------	-------------------------------------

---

**Description**

Setting up a config object with the required details which project to scrape. Also allows for a few arguments forwarded to `httr2` to throttle the HTTP requests (being nice to the server).

**Usage**

```
fs_get_config(
  project,
  title,
  syntaxhighlighting = FALSE,
  timeout = 60,
  capacity = 1,
  filltime = 1
)
```

**Arguments**

<code>project</code>	character, name of the project (e.g., zoo, exams) pointing to an existing R-Forge project.
<code>title</code>	character, title of the project (can be any title).
<code>syntaxhighlighting</code>	logical, defaults to FALSE. If set TRUE all code chunks use R syntax highlighting (though only being 'plain text') but are not evaluated (not run).
<code>timeout</code>	positive numeric, defaults to 60 seconds. Forwarded to <code>httr2::req_timeout()</code> .
<code>capacity</code>	positive numeric, defaults to 1. Forwarded to <code>httr2::req_throttle()</code> .
<code>filltime</code>	positive numeric, defaults to 1. Forwarded to <code>httr2::req_throttle()</code> .

**Value**

Returns an object of class `fs_config` containing the details from a dot-env file as well as a request handler used for scraping the website.

**Author(s)**

Reto

---

*fs\_get\_forum\_details Loading Downloaded Forum Details*

---

**Description**

Whilst downloading (scraping) the content of the forum, the topic/message details are stored in a `forum_{forum_id}` folder. This function loads, checks, and returns all the details to be used to create the Quarto website.

**Usage**

```
fs_get_forum_details(config, verbose = TRUE, n = NULL)
```

**Arguments**

<code>config</code>	an object of class "fs_config" as returned by <code>fs_get_config()</code> .
<code>verbose</code>	logical, produces some output if set TRUE.
<code>n</code>	NULL or positive integer. Just for testing! If integer, a maximum of <code>n</code> threads per forum will be used.

**Details**

Expects two files located in a folder named `forum_{forum_id}`, namely:

If any of the messages comes with an attachment, a folder `forum_{forum_id}/attachments` exist containing the original attachments. Will be checked by this function.

**Value**

Object of class `fs_forum_details`, a named list with three elements, namely `threads` (contains the list of all threads of the forum), `messages` (list of messages, one for each thread), `users` (user profile information) and `config` (a copy of the argument `config`).

**Author(s)**

Reto

---

fs_get_forums	<i>Getting Available Forums</i>
---------------	---------------------------------

---

**Description**

Getting Available Forums

**Usage**

```
fs_get_forums(config, verbose = TRUE)
```

**Arguments**

config	an object of class "fs_config" as returned by <a href="#">fs_get_config()</a> .
verbose	logical, defaults to TRUE.

**Value**

Returns a tibble data frame with all available forums (if any). else an error will be thrown.

**Author(s)**

Reto

---

fs_get_messages	<i>Scraping Messages from Thread/Topic</i>
-----------------	--

---

**Description**

Scraping Messages from Thread/Topic

**Usage**

```
fs_get_messages(threads, config, verbose = TRUE)
```

**Arguments**

threads	object of class fs_thread as returned by <a href="#">fs_get_all_threads()</a> .
config	an object of class "fs_config" as returned by <a href="#">fs_get_config()</a> .
verbose	logical, will procud some output if it evaluates to TRUE.

**Value**

Invisibly returns a list a named list of tibble data frames with all messages for a specific thread, identified by the list names (thread ID). This return is typically not used, as the scraped details are also stored in the RDS files `project_{config$PROJECT_NAME}/thread_{thread_id}.rds` to be used later creating the Quarto markdown files, alongside a `project_{config$PROJECT_NAME}/users.rds` file with user details.

**Author(s)**

Reto

---

fs_write_quarto	<i>Creating Quarto Sources</i>
-----------------	--------------------------------

---

**Description**

Based on the forum details found on disc (see [fs\\_get\\_forum\\_details\(\)](#)) this function creates the quarto sources for the (static) Quarto website.

**Usage**

```
fs_write_quarto(
  x,
  overwrite = FALSE,
  verbose = TRUE,
  forum_title = c(`open-discussion` = "Open Discussion", help = "Help", developers =
    "Developers")
)
```

**Arguments**

x	object of class <code>fs_forum_details</code> as returned by <a href="#">fs_get_forum_details()</a> .
overwrite	logical, defaults to FALSE. If FALSE and any Quarto website target source file exists, an error will be thrown. If set TRUE, the files in the output directory will be overwritten. Use at your own risk.
verbose	logical, produces some output if set TRUE.
forum_title	a named character vector to translate the forum names. The name of the vector corresponds to the 'original' forum name, the value is used for displaying the forum on the Quarto website.

**Author(s)**

Reto

---

`get_attachment_info`     *Extracting Attachment Information*

---

### **Description**

From an attachment link node, all required details are extracted and returned as a named list. This is used to later download the attachment and save locally (see [download\\_attachment\(\)](#)).

### **Usage**

```
get_attachment_info(x)
```

```
get_attachment_info(x)
```

### **Arguments**

x                    an `xml_node` of an attachment.

### **Value**

A named list with the name of the file, the URL for downloading the file later on, as well as the file ID.

Returns a named list with the name of the file, the URL for downloading the file later on, as well as the file ID.

### **Author(s)**

Reto

# Index

`download_attachment`, [2](#)  
`download_attachment()`, [7](#)

`fs_get_all_threads`, [2](#)  
`fs_get_all_threads()`, [5](#)  
`fs_get_config`, [3](#)  
`fs_get_config()`, [2](#), [4](#), [5](#)  
`fs_get_forum_details`, [4](#)  
`fs_get_forum_details()`, [6](#)  
`fs_get_forums`, [5](#)  
`fs_get_forums()`, [2](#)  
`fs_get_messages`, [5](#)  
`fs_write_quarto`, [6](#)

`get_attachment_info`, [7](#)