

# Package ‘conflr’

October 12, 2022

**Type** Package

**Title** Client for 'Confluence' API

**Version** 0.1.1

**Description** Provides utilities for working with various 'Confluence' API  
<<https://docs.atlassian.com/ConfluenceServer/rest/latest/>>, including a  
functionality to convert an R Markdown document to 'Confluence' format and  
upload it to 'Confluence' automatically.

**URL** <https://line.github.io/conflr/>, <https://github.com/line/conflr>

**SystemRequirements** pandoc (>= 1.12.3) - <https://pandoc.org>

**BugReports** <https://github.com/line/conflr/issues>

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Imports** askpass, commonmark, curl, glue, httr, knitr, miniUI, purrr,  
rmarkdown, rstudioapi, shiny, stringi, xml2, R6, rlang

**Suggests** mockery, testthat (>= 2.1.0), withr

**RoxygenNote** 7.1.0

**NeedsCompilation** no

**Author** Hiroaki Yutani [aut, cre],  
LINE Corporation [cph]

**Maintainer** Hiroaki Yutani <[hiroaki.yutani@linecorp.com](mailto:hiroaki.yutani@linecorp.com)>

**Repository** CRAN

**Date/Publication** 2020-04-08 12:50:02 UTC

## R topics documented:

conflr	2
confl_attachment	2
confl_content	4

confl_contentbody . . . . .	5
confl_create_post_from_Rmd . . . . .	6
confl_space . . . . .	8
confl_user . . . . .	9

<b>Index</b>	<b>11</b>
--------------	-----------

---

conflr	<i>R Client for 'Confluence' API</i>
--------	--------------------------------------

---

## Description

Provides utilities for working with various 'Confluence' API <<https://docs.atlassian.com/ConfluenceServer/rest/latest/>>, including a functionality to convert an R Markdown document to 'Confluence' format and upload it to 'Confluence' automatically.

## Author(s)

**Maintainer:** Hiroaki Yutani <[hiroaki.yutani@linecorp.com](mailto:hiroaki.yutani@linecorp.com)>

Other contributors:

- LINE Corporation [copyright holder]

## See Also

Useful links:

- <https://line.github.io/conflr/>
- <https://github.com/line/conflr>
- Report bugs at <https://github.com/line/conflr/issues>

---

confl_attachment	<i>CRUD Operations for Attachments on Content</i>
------------------	---

---

## Description

CRUD Operations for Attachments on Content

**Usage**

```

confl_list_attachments(
  id,
  filename = NULL,
  mediaType = NULL,
  start = 0,
  limit = 50,
  expand = NULL
)

confl_post_attachment(id, path)

confl_update_attachment_metadata(id, attachmentId, ...)

confl_update_attachment_data(id, attachmentId, path, ...)

```

**Arguments**

<code>id</code>	The ID of a page that attachments belong to.
<code>filename</code>	Filter parameter to return only the Attachment with the matching file name. Optional.
<code>mediaType</code>	Filter parameter to return only Attachments with a matching Media-Type. Optional.
<code>start</code>	The start point of the collection to return.
<code>limit</code>	The limit of the number of items to return, this may be restricted by fixed system limits.
<code>expand</code>	A comma separated list of properties to expand. To refer the nested contents, use periods. (e.g. <code>body.storage,history</code> ).
<code>path</code>	Path to a file to upload.
<code>attachmentId</code>	The ID of an attachment.
<code>...</code>	Other arguments passed to 'query'.

**Value**

The API response as a list.

**Examples**

```

## Not run:
# Create a dummy text file
tmp_txt <- tempfile(fileext = ".txt")
cat("foo", file = tmp_txt)

# Upload the file to a page whose ID is "123"
confl_post_attachment("123", tmp_txt)

# Confirm the file is attached to the page

```

```

result <- confl_list_attachments("123", filename = basename(tmp_txt))
length(result$results) # should be 1

## End(Not run)

```

---

confl\_content

*REST Wrapper for the ContentService*


---

## Description

REST Wrapper for the ContentService

## Usage

```

confl_list_pages(
  type = c("page", "blogpost", "comment", "attachment"),
  limit = 10,
  start = 0,
  spaceKey = NULL,
  title = NULL,
  expand = NULL
)

```

```

confl_get_page(id, expand = "body.storage")

```

```

confl_post_page(
  type = c("page", "blogpost"),
  spaceKey,
  title,
  body,
  ancestors = NULL
)

```

```

confl_update_page(id, title, body)

```

```

confl_delete_page(id)

```

## Arguments

type	The content type to return. Default value: page. Valid values: page, blogpost.
limit	The limit of the number of items to return, this may be restricted by fixed system limits.
start	The start point of the collection to return.
spaceKey	The space key to find content under.
title	The title of the page to find. Required for page type.

expand	A comma separated list of properties to expand. To refer the nested contents, use periods. (e.g. body.storage,history).
id	ID of the content.
body	The HTML source of the page.
ancestors	The page ID of the parent pages.

**Value**

The API response as a list.

**See Also**

<https://docs.atlassian.com/ConfluenceServer/rest/latest/>

**Examples**

```
## Not run:
# Create a page titled "title1" on a space named "space1"
result <- confl_post_page(
  type = "page",
  spaceKey = "space1",
  title = "title1",
  body = "<h2>example</h2><p>This is example</p>"
)

# Jump to the result page
browseURL(paste0(result$`_links`$base, result$`_links`$webui))

# List pages under space "space1" up to 10 pages
confl_list_pages(spaceKey = "space1")

## End(Not run)
```

---

confl\_contentbody      *Converts between content body representations*

---

**Description**

Converts between content body representations

**Usage**

```
confl_contentbody_convert(
  x,
  from = c("wiki", "storage", "editor", "view", "export_view", "styled_view"),
  to = c("storage", "editor", "view", "export_view", "styled_view")
)
```

**Arguments**

x	The content body to convert.
from	The format to convert from.
to	The format to convert to.

**Value**

The API response as a list.

**See Also**

<https://docs.atlassian.com/ConfluenceServer/rest/latest/>

**Examples**

```
## Not run:
# Convert to a Math macro
confl_contentbody_convert("\\[1+1=2\\]")

# Convert to an Expand macro
confl_contentbody_convert("{expand}detail is here {expand}")

## End(Not run)
```

---

```
confl_create_post_from_Rmd
Publish R Markdown Document to 'Confluence'
```

---

**Description**

Knit and post a given R Markdown file to 'Confluence'.

**Usage**

```
confl_create_post_from_Rmd(Rmd_file, interactive = NULL, params = NULL, ...)

confluence_document(
  title = NULL,
  space_key = NULL,
  parent_id = NULL,
  type = c("page", "blogpost"),
  toc = FALSE,
  toc_depth = 7,
  code_folding = c("none", "hide"),
  supported_syntax_highlighting = getOption("conflr_supported_syntax_highlighting"),
  update = NULL,
```

```

    use_original_size = FALSE,
    interactive = NULL
  )

```

### Arguments

Rmd_file	Path to an .Rmd file.
interactive	If FALSE, shiny interface is not launched.
params	If provided, a list of named parameters that override custom params in the YAML front-matter.
...	Arguments passed to <code>confluence_documents()</code> .
title	Title of the post.
space_key	The space key to find content under.
parent_id	The page ID of the parent pages.
type	The content type to return. Default value: page. Valid values: page, blogpost.
toc	If TRUE, include a table of contents in the output.
toc_depth	The max level of headers to include in the table of contents.
code_folding	If "hide", fold code blocks by default.
supported_syntax_highlighting	A named character vector of supported syntax highlighting other than default (e.g. <code>c(r = "r")</code> ).
update	If TRUE, overwrite the existing page (if it exists).
use_original_size	If TRUE, use the original image sizes.

### Details

All options of `confluence_document()` can also be specified via the argument of `confl_create_post_from_Rmd`. If an option is specified on both, the one given as an argument will be used.

```

---
title: "title1"
output:
  confluence_document:
    space_key: "space1"
    parent_id: 1234
    toc: TRUE
    toc_depth: 4
    code_folding: hide
    supported_syntax_highlighting:
      r: r
      foo: bar
    update: true
    use_original_size: true
---
...

```

**Value**

confl\_create\_post\_from\_Rmd() returns the URL of the published page.  
 confluence\_document() returns an rmarkdown\_output\_format object.

**Examples**

```
example_Rmd <- system.file("extdata/example.Rmd", package = "conflr")

## Not run:
# Convert an R Markdown document into a 'Confluence' page interactively
confl_create_post_from_Rmd(example_Rmd)

# You can override most of the parameters of confluence_document()
confl_create_post_from_Rmd(example_Rmd, space = "space1", toc = TRUE)

## End(Not run)

## Not run:
# A custom R markdown format that can be passed to rmarkdown::render()
format <- confluence_document(space_key = "space1")
rmarkdown::render(system.file("extdata/example.Rmd", package = "conflr"), format)

## End(Not run)
```

---

 confl\_space

*REST Wrapper for the SpaceService*


---

**Description**

REST Wrapper for the SpaceService

**Usage**

```
confl_list_spaces(
  spaceKey = NULL,
  type = c("global", "personal"),
  status = c("current", "archived"),
  label = NULL,
  favourite = NULL,
  expand = NULL,
  start = NULL,
  limit = 25
)

confl_get_space(spaceKey, expand = NULL)
```



**Arguments**

spaceKey	The space key to find content under.
type	Filter the list of spaces returned by type (global, personal).
status	Filter the list of spaces returned by status (current, archived).
label	Filter the list of spaces returned by label.
favourite	Filter the list of spaces returned by favourites.
expand	A comma separated list of properties to expand. To refer the nested contents, use periods. (e.g. body.storage,history).
start	The start point of the collection to return.
limit	The limit of the number of items to return, this may be restricted by fixed system limits.

**Value**

The API response as a list.

**Examples**

```
## Not run:
# Get the information of a space named "space1"
confl_get_space("space1")

## End(Not run)
```

---

confl\_user

*Non-admin User Operations*

---

**Description**

Non-admin User Operations

**Usage**

```
confl_get_user(key = NULL, username = NULL, expand = NULL)
```

```
confl_get_current_user(expand = NULL)
```

**Arguments**

key	Userkey of the user to request from this resource.
username	Username of the user to request from this resource.
expand	A comma separated list of properties to expand. To refer the nested contents, use periods. (e.g. body.storage,history).

**Value**

The API response as a list.

**Examples**

```
## Not run:  
# Get the information of the current user  
my_user <- confl_get_current_user()  
  
# Show display name  
my_user$displayName  
  
# Get the information of a user whose name is "user1"  
other_user <- confl_get_user(username = "user1")  
  
## End(Not run)
```

# Index

- confl\_attachment, 2
- confl\_content, 4
- confl\_contentbody, 5
- confl\_contentbody\_convert
  - (confl\_contentbody), 5
- confl\_create\_post\_from\_Rmd, 6
- confl\_delete\_page (confl\_content), 4
- confl\_get\_current\_user (confl\_user), 9
- confl\_get\_page (confl\_content), 4
- confl\_get\_space (confl\_space), 8
- confl\_get\_user (confl\_user), 9
- confl\_list\_attachments
  - (confl\_attachment), 2
- confl\_list\_pages (confl\_content), 4
- confl\_list\_spaces (confl\_space), 8
- confl\_post\_attachment
  - (confl\_attachment), 2
- confl\_post\_page (confl\_content), 4
- confl\_space, 8
- confl\_update\_attachment\_data
  - (confl\_attachment), 2
- confl\_update\_attachment\_metadata
  - (confl\_attachment), 2
- confl\_update\_page (confl\_content), 4
- confl\_user, 9
- conflr, 2
- conflr-package (conflr), 2
- confluence\_document
  - (confl\_create\_post\_from\_Rmd), 6