

# Package ‘jinjar’

May 1, 2022

**Title** Template Engine Inspired by 'Jinja'

**Version** 0.1.1

**Description** Template engine powered by the 'inja' C++ library. Users write a template document, using syntax inspired by the 'Jinja' Python package, and then render the final document by passing data from R. The template syntax supports features such as variables, loops, conditions and inheritance.

**License** MIT + file LICENSE

**URL** <https://davidchall.github.io/jinjar/>,  
<https://github.com/davidchall/jinjar>

**BugReports** <https://github.com/davidchall/jinjar/issues>

**Imports** checkmate, fs, jsonlite, rlang

**Suggests** knitr, rmarkdown, spelling, testthat (>= 3.0.0)

**LinkingTo** cpp11

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**Encoding** UTF-8

**Language** en-US

**RoxygenNote** 7.1.2

**SystemRequirements** C++11

**NeedsCompilation** yes

**Author** David Hall [aut, cre, cph] (<<https://orcid.org/0000-0002-2193-0480>>),  
Lars Berscheid [cph] (Author of bundled inja library),  
Niels Lohmann [cph] (Author of bundled json library)

**Maintainer** David Hall <david.hall.physics@gmail.com>

**Repository** CRAN

**Date/Publication** 2022-05-01 05:10:02 UTC

## R topics documented:

jinjar_config . . . . .	2
loader . . . . .	3
render . . . . .	4

<b>Index</b>	<b>6</b>
--------------	----------

---

jinjar_config	<i>Configure the templating engine</i>
---------------	--

---

### Description

Create an object to configure the templating engine behavior (e.g. customize the syntax). The default values have been chosen to match the Jinja defaults.

### Usage

```
jinjar_config(
  loader = NULL,
  block_open = "{%",
  block_close = "%}",
  variable_open = "{{",
  variable_close = "}}",
  comment_open = "{#",
  comment_close = "#}",
  line_statement = NULL,
  trim_blocks = FALSE,
  lstrip_blocks = FALSE,
  ignore_missing_files = FALSE
)

default_config()
```

### Arguments

**loader** How the engine discovers templates. Choices:

- NULL (default), disables search for templates.
- Path to template directory.
- A [loader](#) object.

**block\_open, block\_close** The opening and closing delimiters for control blocks. Default: "{%" and "%}".

**variable\_open, variable\_close** The opening and closing delimiters for print statements. Default: "{{" and "}}".

**comment\_open, comment\_close** The opening and closing delimiters for comments. Default: "{#" and "#}".

`line_statement` The prefix for an inline statement. If NULL (the default), inline statements are disabled.

`trim_blocks` Remove first newline after a block. Default: FALSE.

`lstrip_blocks` Remove inline whitespace before a block. Default: FALSE.

`ignore_missing_files` Ignore include or extends statements when the auxiliary template cannot be found. If FALSE (default), then an error is raised.

**Value**

A "jinjar\_config" object.

**Note**

The equivalent Jinja class is `Environment`, but this term has special significance in R (see [environment\(\)](#)).

**Examples**

```
jinjar_config()
```

---

loader *Template loaders*

---

**Description**

Loaders are responsible for exposing templates to the templating engine.

`path_loader()` loads templates from a directory in the file system.

`package_loader()` loads templates from a directory in an R package.

`list_loader()` loads templates from a named list.

**Usage**

```
path_loader(...)
```

```
package_loader(package, ...)
```

```
list_loader(x)
```

**Arguments**

`...` Strings specifying path components.

`package` Name of the package in which to search.

`x` Named list mapping template names to template sources.

**Value**

A "jinja\_loader" object.

**See Also**

The loader is an argument to [jinja\\_config\(\)](#).

**Examples**

```
path_loader(R.home())

package_loader("base", "demo")

list_loader(list(
  header = "Title: {{ title }}",
  content = "Hello {{ person }}!"
))
```

---

render

*Render template*

---

**Description**

Data is passed to a template to render the final document.

**Usage**

```
render(.x, ...)
```

## S3 method for class 'character'

```
render(.x, ..., .config = default_config())
```

## S3 method for class 'fs\_path'

```
render(.x, ..., .config = default_config())
```

**Arguments**

<code>.x</code>	The template. Choices: <ul style="list-style-type: none"> <li>• A template string.</li> <li>• A path to a template file (use <a href="#">fs::path()</a>).</li> </ul>
<code>...</code>	<dynamic-dots> Data passed to the template.
<code>.config</code>	The engine configuration. The default matches Jinja defaults, but you can use <a href="#">jinja_config()</a> to customize things like syntax delimiters, whitespace control, and loading auxiliary templates.

**Value**

String containing rendered template.

**See Also**

`vignette("template-syntax")` describes how to write templates and render them using data variables.

**Examples**

```
# pass data as arguments
render("Hello {{ name }}!", name = "world")

# pass data as list
params <- list(name = "world")
render("Hello {{ name }}!", !!!params)

# render template file
## Not run:
render(fs::path("template.txt"), name = "world")

## End(Not run)
```

# Index

`default_config(jinjar_config)`, 2

`environment()`, 3

`fs::path()`, 4

`jinjar_config`, 2

`jinjar_config()`, 4

`list_loader(loader)`, 3

`loader`, 2, 3

`package_loader(loader)`, 3

`path_loader(loader)`, 3

`render`, 4