

Package ‘rjade’

April 16, 2021

Type Package

Title A Clean, Whitespace-Sensitive Template Language for Writing HTML

Version 0.1.1

Author Jeroen Ooms, Forbes Lindesay

Maintainer Jeroen Ooms <jeroen@berkeley.edu>

Description Jade is a high performance template engine heavily influenced by Haml and implemented with JavaScript for node and browsers.

License MIT + file LICENSE

URL <https://github.com/jeroen/rjade> <https://www.npmjs.com/package/jade>

BugReports <https://github.com/jeroen/rjade/issues>

VignetteBuilder knitr

Imports V8 (>= 0.5)

Suggests knitr, rmarkdown

RoxygenNote 7.1.1

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2021-04-16 13:00:02 UTC

R topics documented:

jade_compile	2
Index	3

`jade_compile`*Render Jade Template*

Description

Jade is a high performance template engine heavily influenced by Haml.

Usage

```
jade_compile(text, ...)
```

```
jade_render(text, ..., locals = list())
```

Arguments

`text` string with jade template.

`...` options passed to the compiler, see <https://jade-lang.com/api>.

`locals` local variables used in the template.

Details

Converting a template to HTML text involves two steps. The first step compiles the template with some formatting options into a closure. The binding for this is implemented in `jade_compile`. The second step calls the closure with optionally some local variables to render the output to HTML.

The `jade_render` function is a convenience wrapper that does both steps at once. This is slightly faster if you only need to render your template once.

References

Jade documentation: <https://jade-lang.com>

Examples

```
# Example from https://jade-lang.com
text <- readLines(system.file("examples/test.jade", package = "rjade"))
```

```
# Compile and render seperately
tpl <- jade_compile(text, pretty = TRUE)
tpl()
tpl(youAreUsingJade = TRUE)
```

```
# Slightly faster for one-time rendering
jade_render(text, pretty = TRUE)
jade_render(text, pretty = TRUE, locals = list(youAreUsingJade = TRUE))
```

Index

`jade (jade_compile)`, [2](#)
`jade_compile`, [2](#)
`jade_render (jade_compile)`, [2](#)