

Package ‘jug’

April 13, 2017

Type Package

Title A Simple Web Framework for R

Version 0.1.7

URL <https://github.com/Bart6114/jug>

Description

jug is a web framework aimed at easily building APIs. It is mostly aimed at exposing R functions, models and visualizations to third-parties by way of http requests.

License MIT + file LICENSE

LazyData TRUE

Depends R (>= 3.1.0), magrittr

Imports R6, webutils, httpuv, mime, infuser, jsonlite, base64enc

Suggests testthat, knitr, rmarkdown

VignetteBuilder knitr

RoxygenNote 6.0.1

NeedsCompilation no

Author Bart Smeets [aut, cre]

Maintainer Bart Smeets <bartsmeets86@gmail.com>

Repository CRAN

Date/Publication 2017-04-13 17:51:48 UTC

R topics documented:

add_middleware	2
auth_basic	3
collector	3
cors	3
decorate	4
delete	4
Error	5
get	5

include	6
jug	6
match_path	7
Middleware	7
MiddlewareHandler	8
new_error	8
parse_params	8
post	9
process_test_request	9
put	10
RawTestRequest	10
Request	11
Response	11
serve_it	11
serve_static_files	12
simple_error_handler	12
simple_error_handler_json	13
stop_daemon	13
use	14
ws	14

Index 15

add_middleware	<i>Internal function to add middleware</i>
----------------	--

Description

Internal function to add middleware

Usage

```
add_middleware(jug, func, path = NULL, method = NULL, websocket = FALSE)
```

Arguments

jug	the jug instance
func	the function to bind
path	the path to bind to
method	the method to bind to
websocket	should the middleware bind to the websocket protocol

auth_basic	<i>Basic authentication middleware</i>
------------	--

Description

Basic authentication middleware

Usage

```
auth_basic(account_eval_func, basic_realm = "this_jug_server")
```

Arguments

account_eval_func	a function to which the username and password arguments will be passed and which should return TRUE for a valid combination and FALSE for an invalid one.
basic_realm	the user visible realm that will be returned through the WWW-Authenticate header in case of an unauthenticated request

collector	<i>Collector function for middlewares</i>
-----------	---

Description

Basically constructs an empty jug instances that will be merged into the primary one later on

Usage

```
collector()
```

cors	<i>Add cors functionality</i>
------	-------------------------------

Description

Allows to set CORS-headers as specified in https://developer.mozilla.org/en-US/docs/Web/HTTP/Access_control_CORS. It will also setup the infrastructure in order to receive OPTIONS requests send to path.

Usage

```
cors(jug, path = NULL, allow_methods = c("POST", "GET", "PUT", "OPTIONS",
    "DELETE", "PATCH"), allow_origin = "*", allow_credentials = NULL,
    allow_headers = NULL, max_age = NULL, expose_headers = NULL)
```

Arguments

jug	the jug instance
path	the path to bind to
allow_methods	a vector of allowed methods
allow_origin	the allowed origin
allow_credentials	either true or false (supply as string)
allow_headers	a vector of allowed headers
max_age	in seconds
expose_headers	a vector of exposed headers

decorate	<i>Convenience function to decorate existing functions</i>
----------	--

Description

Convenience function to decorate existing functions

Usage

```
decorate(func, content_type = "text/html", strict_params = FALSE)
```

Arguments

func	the function to decorate
content_type	the content type to set on the response object
strict_params	if TRUE and not all parameters needed for func

delete	<i>Function to add DELETE-binding middleware</i>
--------	--

Description

Function to add DELETE-binding middleware

Usage

```
delete(jug, path, ...)
```

Arguments

jug	the jug object
path	the path to bind to
...	functions (order matters) to bind to the path (will receive the params req, res and err)

See Also

[post](#), [put](#), [get](#), [use](#), [ws](#)

Error	<i>Response class</i>
-------	-----------------------

Description

Response class

Usage

Error

Format

An object of class R6ClassGenerator of length 24.

get	<i>Function to add GET-binding middleware</i>
-----	---

Description

The params to this function are slightly different from post, put and the like due to get being a base function.

Usage

```
get(object, ...)

## S3 method for class 'Jug'
get(object, ...)
```

Arguments

object	the jug object
...	first argument should be the path followed by middleware functions (order matters) to bind to the path (will receive the params req, res and err)

Methods (by class)

- Jug: Function to add GET-binding middleware

See Also

[post](#), [put](#), [delete](#), [use](#), [ws](#)

include	<i>Include elsewhere constructed middleware with primary jug instance</i>
---------	---

Description

Include elsewhere constructed middleware with primary jug instance

Usage

```
include(jug, collector, source_file = NULL)
```

Arguments

jug	the primary jug instance
collector	the variable containing the external middleware
source_file	provide (relative) path if collector variable is located in another .R file

jug	<i>New jug instance</i>
-----	-------------------------

Description

Creates a new jug instance which can be build upon with other functions (middlewares).

Usage

```
jug()
```

Examples

```
# This Hello World example will serve a jug instance on the default port.
# The jug instance will return "Hello World!" if a GET request is send to it.
jug() %>%
  get("/", function(req, res, err) "Hello World!" ) %>%
  simple_error_handler_json() %>%
  serve_it(verbose=TRUE)

# Introduction to jug
vignette("jug", package="jug")
```

 match_path

Helper function to extract regex named capture groups from string

Description

adapted from: <http://oddhypothesis.blogspot.be/2012/05/regex-named-capture-in-r.html>

Usage

```
match_path(pattern, path, ...)
```

Arguments

pattern	regex pattern
path	the path to test the regex pattern against
...	arguments passed to regexpr

 Middleware

Middleware class

Description

Middleware class

Usage

```
Middleware
```

Format

An object of class R6ClassGenerator of length 24.

MiddlewareHandler *MiddlewareHandler R6 class definition*

Description

MiddlewareHandler R6 class definition

Usage

MiddlewareHandler

Format

An object of class R6ClassGenerator of length 24.

new_error *Create response instance*

Description

Create response instance

Usage

new_error()

parse_params *Parse the params passed by the request*

Description

Parse the params passed by the request

Usage

parse_params(req, body, query_string, content_type)

Arguments

req	the rook req environment
body	the parsed body
query_string	a query string
content_type	the mime type

post	<i>Function to add POST-binding middleware</i>
------	--

Description

Function to add POST-binding middleware

Usage

```
post(jug, path, ...)
```

Arguments

jug	the jug object
path	the path to bind to
...	functions (order matters) to bind to the path (will receive the params req, res and err)

See Also

[get](#), [put](#), [delete](#), [use](#), [ws](#)

process_test_request	<i>Initialize process of test request</i>
----------------------	---

Description

Initialize process of test request

Usage

```
process_test_request(jug, test_request)
```

Arguments

jug	the jug instance
test_request	the RawTestRequest instance

put	<i>Function to add PUT-binding middleware</i>
-----	---

Description

Function to add PUT-binding middleware

Usage

```
put(jug, path, ...)
```

Arguments

jug	the jug object
path	the path to bind to
...	functions (order matters) to bind to the path (will receive the params req, res and err)

See Also

[post](#), [get](#), [delete](#), [use](#), [ws](#)

RawTestRequest	<i>Generate request for testing purposes</i>
----------------	--

Description

solely used for testing purposes

Usage

```
RawTestRequest
```

Format

An object of class R6ClassGenerator of length 24.

Request	<i>Request class</i>
---------	----------------------

Description

Request class

Usage

Request

Format

An object of class R6ClassGenerator of length 24.

Response	<i>Response class</i>
----------	-----------------------

Description

Response class

Usage

Response

Format

An object of class R6ClassGenerator of length 24.

serve_it	<i>Start serving the jug instance</i>
----------	---------------------------------------

Description

Start serving the jug instance

Usage

```
serve_it(jug, host = "127.0.0.1", port = 8080, daemonized = FALSE,  
         verbose = FALSE)
```

Arguments

jug	the jug instance
host	the host address
port	the port to host on
daemonized	whether or not to start a daemonized server (experimental)
verbose	verbose output?

serve_static_files *Middleware to serve static files*

Description

Binds to get requests that aren't handled by specified paths. Should support all filetypes; returns image and octet-stream types as a raw string.

Usage

```
serve_static_files(jug, path = NULL, root_path = getwd())
```

Arguments

jug	the jug instance
path	the path to bind to, default = NULL (all paths)
root_path	the file path to set as root for the file server

simple_error_handler *An error handler middleware which returns a error description in HTML format*

Description

An error handler middleware which returns a error description in HTML format

Usage

```
simple_error_handler(jug, path = NULL, to_cat_or_not_to_cat = "cat")
```

Arguments

jug	the jug instance
path	the path to bind to, default = NULL (all paths)
to_cat_or_not_to_cat	the argument says it all

See Also

simple_error_handler_json

simple_error_handler_json

An error handler middleware which returns a error description in JSON format

Description

An error handler middleware which returns a error description in JSON format

Usage

```
simple_error_handler_json(jug, path = NULL)
```

Arguments

jug	the jug instance
path	the path to bind to, default = NULL (all paths)

See Also

simple_error_handler

stop_daemon

Stop daemonized server

Description

Stop daemonized server

Usage

```
stop_daemon(jug)
```

Arguments

jug	the jug instance
-----	------------------

use *Function to add request method insensitive middleware*

Description

Function to add request method insensitive middleware

Usage

```
use(jug, path, ..., method = NULL)
```

Arguments

jug	the jug object
path	the path to bind to
...	functions (order matters) to bind to the path (will receive the params req, res and err)
method	the method to bind to, defaults to NULL

See Also

[post](#), [put](#), [delete](#), [get](#), [ws](#)

ws *Function to add websocket handling middleware*

Description

Function to add websocket handling middleware

Usage

```
ws(jug, path, ...)
```

Arguments

jug	the jug object
path	the path to bind to
...	functions (order matters) to bind to the path (will receive the params req, res and err)

See Also

[post](#), [put](#), [delete](#), [get](#), [get](#)

Index

*Topic **datasets**

- Error, [5](#)
 - Middleware, [7](#)
 - MiddlewareHandler, [8](#)
 - RawTestRequest, [10](#)
 - Request, [11](#)
 - Response, [11](#)
- [add_middleware](#), [2](#)
- [auth_basic](#), [3](#)
- [collector](#), [3](#)
- [cors](#), [3](#)
- [decorate](#), [4](#)
- [delete](#), [4](#), [6](#), [9](#), [10](#), [14](#)
- [Error](#), [5](#)
- [get](#), [5](#), [5](#), [9](#), [10](#), [14](#)
- [include](#), [6](#)
- [jug](#), [6](#)
- [match_path](#), [7](#)
- [Middleware](#), [7](#)
- [MiddlewareHandler](#), [8](#)
- [new_error](#), [8](#)
- [parse_params](#), [8](#)
- [post](#), [5](#), [6](#), [9](#), [10](#), [14](#)
- [process_test_request](#), [9](#)
- [put](#), [5](#), [6](#), [9](#), [10](#), [14](#)
- [RawTestRequest](#), [10](#)
- [Request](#), [11](#)
- [Response](#), [11](#)
- [serve_it](#), [11](#)
- [serve_static_files](#), [12](#)
- [simple_error_handler](#), [12](#)
- [simple_error_handler_json](#), [13](#)
- [stop_daemon](#), [13](#)
- [use](#), [5](#), [6](#), [9](#), [10](#), [14](#)
- [ws](#), [5](#), [6](#), [9](#), [10](#), [14](#), [14](#)