Package ‘tower’

October 21, 2024

Title Easy Middle Ware Library for 'shiny'

Version 0.2.0

Description The best way to implement middle ware for 'shiny' Applications. 'tower' is designed to make implementing behavior on top of 'shiny' easy with a layering model for incoming HTTP requests and server sessions. 'tower' is a very minimal package with little overhead, it is mainly meant for other package developers to implement new behavior.

URL `https://github.com/ixpantia/tower`

BugReports `https://github.com/ixpantia/tower/issues`

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Imports compiler, glue, purrr, stringr, curl, jsonlite, rlang

Suggests testthat (>= 3.0.0), shiny

Config/testthat/edition 3

NeedsCompilation no

Author ixpantia, SRL [cph],
Andres Quintero [aut, cre]

Maintainer Andres Quintero <`andres@ixpantia.com`>

Repository CRAN

Date/Publication 2024-10-21 11:40:08 UTC

Contents

add_body *Add a body to a response*

Description

Adds a body to a response, if no content type is set, it will be detected

Usage

```
add_body(res, body)
```

Arguments

`res`	A response builder object
`body`	The body to add

Value

The response builder object

`add_body_json` *Add a body to a response as JSON*

Description

Adds a body to a response as JSON

Usage

```
add_body_json(res, body)
```

Arguments

`res`	A response builder object
`body`	The body to add

Value

The response builder object

`add_cookie` *Add a cookie to a response*

Description

Adds a cookie to a response

Usage

```
add_cookie(res, name, value)
```

Arguments

`res`	A response builder object
`name`	The name of the cookie
`value`	The value of the cookie

Value

The response builder object

`add_delete_route` *Add a DELETE route*

Description

Adds a DELETE route to a tower

Usage

```
add_delete_route(tower, path, handler)
```

Arguments

`tower`	A tower object
`path`	A string containing the path to match
`handler`	A function to call when the route is matched

Value

The tower with the added DELETE route

`add_get_route` *Add a GET route*

Description

Adds a GET route to a tower

Usage

```
add_get_route(tower, path, handler)
```

Arguments

`tower`	A tower object
`path`	A string containing the path to match
`handler`	A function to call when the route is matched

Value

The tower with the added GET route

add_http_layer — *Add an HTTP layer to a tower*

Description

Add an HTTP layer to a tower. This layer will be called before the 'shiny' app's httpHandler.

Usage

```
add_http_layer(tower, layer)
```

Arguments

`tower`	A tower
`layer`	A function that takes a request and returns either a response. A layer can short circuit by returning a response directly or call the next layer will `req$NEXT(req)` which will call the next layer in the middleware.

Value

The tower with the added layer

add_patch_route — *Add a PATCH route*

Description

Adds a PATCH route to a tower

Usage

```
add_patch_route(tower, path, handler)
```

Arguments

`tower`	A tower object
`path`	A string containing the path to match
`handler`	A function to call when the route is matched

Value

The tower with the added PATCH route

`add_post_route` *Add a POST route*

Description

Adds a POST route to a tower

Usage

```
add_post_route(tower, path, handler)
```

Arguments

`tower`	A tower object
`path`	A string containing the path to match
`handler`	A function to call when the route is matched

Value

The tower with the added POST route

`add_put_route` *Add a PUT route*

Description

Adds a PUT route to a tower

Usage

```
add_put_route(tower, path, handler)
```

Arguments

`tower`	A tower object
`path`	A string containing the path to match
`handler`	A function to call when the route is matched

Value

The tower with the added PUT route

`add_route` *Add an HTTP layer to a tower*

Description

Adds an HTTP layer to a tower

Usage

```
add_route(tower, method = "GET", path, handler)
```

Arguments

`tower`	A tower object
`method`	A string containing the HTTP method to match
`path`	A string containing the path to match
`handler`	A function to call when the layer is matched

Value

The tower with the added route

`add_server_layer` *Add a server layer to a tower*

Description

Add a server layer to a tower. This layer will run before the 'shiny' app's server function. This is useful for adding custom logic to the server function without modifying the original server function.

Usage

```
add_server_layer(tower, layer)
```

Arguments

`tower`	A tower
`layer`	A function that takes input, output, and session and has no return value. This function will be called before the original server function. If you want to short-circuit the server use an exception.

Value

The tower with the added layer

`app_into_parts` *Into parts*

Description

Splits a shiny.appobj into its parts, the ui and server

Usage

```
app_into_parts(app)
```

Arguments

`app`	A shiny.appobj

Value

A list with the ui and server handlers

`build_http_cookie` *Build a cookie*

Description

Builds an HttpOnly cookie from a key and value

Usage

```
build_http_cookie(key, value)
```

Arguments

`key`	A string containing the cookie key
`value`	A string containing the cookie value

Value

A string containing the formated cookie

`build_response` *Build a response*

Description

Builds a response

Usage

```
build_response(res)
```

Arguments

`res`	A response builder object

Value

A 'shiny' response object

`build_tower` *Build a 'shiny' app from a tower*

Description

Build a 'shiny' app from a tower. This will create a new 'shiny' app with the specified layers added.

Usage

```
build_tower(tower)
```

Arguments

`tower`	A tower

Value

A 'shiny' app object that can be started

`create_tower` *Create a new tower*

Description

Create a new tower to build upon.

Usage

```
create_tower(app)
```

Arguments

`app`	A 'shiny' app object

Value

A new tower object to add more layers to

`print.tower` *Print a tower*

Description

Print a tower

Usage

```
## S3 method for class 'tower'
print(x, ...)
```

Arguments

`x`	A tower
`...`	Ignored arguments (for compatibility with print)

Value

No return value, called for side effects

`req_body_form` *Extract form data from a request*

Description

Extracts form data from a request

Usage

```
req_body_form(req)
```

Arguments

`req`	A request object

Value

A list containing the form data in the body

`req_body_json` *Extract the request body from a JSON request*

Description

Extracts the request body from a JSON request

Usage

```
req_body_json(req, ...)
```

Arguments

`req`	A request object
`...`	Additional arguments to pass to `fromJSON` when parsing the request body. This will only be used the first time the request body is parsed. Subsequent calls will return the cached result.

Value

The R object representation of the body's JSON content

`req_cookies` *Extract cookies from a request*

Description

Extracts cookies from a request

Usage

```
req_cookies(req)
```

Arguments

`req`	A request object

Value

A list containing the cookies

`req_query` *Extract query parameters from a request*

Description

Extracts query parameters from a request

Usage

```
req_query(req)
```

Arguments

`req`	A request object

Value

A list containing the query parameters

`response_builder` *Create a response builder*

Description

Creates a response builder

Usage

```
response_builder()
```

Value

A response builder object

`set_content_type` *Set the content type of a response*

Description

Sets the content type of a response

Usage

```
set_content_type(res, content_type)
```

Arguments

`res`	A response builder object
`content_type`	The content type to set

Value

The response builder object

`set_header` *Set a header on a response*

Description

Sets or adds a header to a response

Usage

```
set_header(res, name, value)
```

Arguments

`res`	A response builder object
`name`	The name of the header
`value`	The value of the header

Value

The response builder object

`set_status` *Set the status of a response*

Description

Sets the status of a response

Usage

```
set_status(res, status)
```

Arguments

`res`	A response builder object
`status`	The status to set

Value

The response builder object

Index