

Package: struct (via r-universe)

October 10, 2026

Title Strict modification of objects

Version 0.0.0.9000

Description What the package does (one paragraph).

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Imports purrr, rlang, tibble, vctrs

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

URL <https://github.com/cynkra/struct>

BugReports <https://github.com/cynkra/struct/issues>

Repository <https://cynkra.r-universe.dev>

Date/Publication 2025-06-03 17:33:55 UTC

RemoteUrl <https://github.com/cynkra/struct>

RemoteRef HEAD

RemoteSha a8c64e37ab5276a7fcb2e04aced0ba3f79a3c30d

Contents

bind_structs	2
list_struct	2
print_tree	3
Index	4

bind_structs	<i>bind objects into a tibble_struct</i>
--------------	--

Description

bind objects into a tibble_struct

Usage

```
bind_structs(...)
```

list_struct	<i>struct classes</i>
-------------	-----------------------

Description

Functions to convert to and from struct classes or to test objects.

Usage

```
list_struct(...)
```

```
as_list_struct(x)
```

```
un_list_struct(x)
```

```
is_list_struct(x)
```

```
tibble_struct(...)
```

```
as_tibble_struct(x)
```

```
un_tibble_struct(x)
```

```
is_tibble_struct(x)
```

```
scalar(x)
```

```
un_scalar(x)
```

```
is_scalar(x)
```

Arguments

...	elements of the object to build
x	object

`print_tree`*print tree structure of object*

Description

Not quite right yet. We want to be able to check easily the structure of an object. Doesn't dive deep into lists unless they are named. Ultimately should dive into the list_of ptypes though, and look prettier.

Usage

```
print_tree(x)
```

Arguments

x	object
---	--------

Index

`as_list_struct(list_struct)`, [2](#)
`as_tibble_struct(list_struct)`, [2](#)

`bind_structs`, [2](#)

`is_list_struct(list_struct)`, [2](#)
`is_scalar(list_struct)`, [2](#)
`is_tibble_struct(list_struct)`, [2](#)

`list_struct`, [2](#)

`print_tree`, [3](#)

`scalar(list_struct)`, [2](#)

`tibble_struct(list_struct)`, [2](#)

`un_list_struct(list_struct)`, [2](#)
`un_scalar(list_struct)`, [2](#)
`un_tibble_struct(list_struct)`, [2](#)