Package: tinylens (via r-universe)

November 14, 2024

Title Minimal implementation of functional lenses

Version 0.0.0.9009

Description Minimal implementation of functional lenses, inspired by the `lenses` package.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports rlang, S7, tidyselect, vctrs

Collate 'lens.R' 'verbs.R' 'base-lenses.R' 'dataframe-lenses.R' 'tinylens-package.R' 'zzz.R'

Suggests tinytest

Repository https://arbelt.r-universe.dev

RemoteUrl https://github.com/arbelt/tinylens

RemoteRef HEAD

RemoteSha e31d8741c09433300aa71c3394d7116b426dc9e7

Contents

`attr_l` *Attributes lens*

Description

Lens into a named attribute of an object.

Usage

```
attr_l(name)
```

Arguments

`name`	Name of the attribute to lens into

Value

A lens that selects the specified attribute

`c_l` *Lens for accessing and modifying nested elements of a list or vector*

Description

Convenience function that mirrors `purrr::pluck()`.

Usage

```
c_l(...)
```

Arguments

`...`	A sequence of lenses and/or integers/logical vectors

filter_il *Filter ilens*

Description

This function returns an illegal lens that filters according to the specified conditions.

Usage

```
filter_il(...)
```

Arguments

...	Conditions to filter by

Details

Conditions are evaluated in the context of the data frame.

Value

A lens that filters the specified rows

id_l *Identity lens*

Description

Trivial identity lens: returns and sets the object itself.

Usage

```
id_l
```

Format

An object of class `tinylens::lens` (inherits from `S7_object`) of length 1.

index_l *Index lens*

Description

Lens into a single element of a list.

Usage

```
index_l(i)
```

Arguments

i	Index of the element to lens into

Details

This lens performs indexing using double bracket notation, i.e., `x[[i]]`.

Value

A lens that selects the specified element

lens *Create a lens*

Description

A lens is a pair of functions that can be used to view and set a value in an object.

Usage

```
lens(view = class_missing, set = class_missing)
```

Arguments

view	A function that takes an object and returns a value
set	A function that takes an object and a value and returns a new object

map_l *Lens into a list or vector*

Description

This lens allows you to access and modify elements of a list or vector based on their position or a logical condition.

Usage

```
map_l(l, .ptype = NULL)

map_df_l(l)
```

Arguments

l	A lens that selects the elements to lens into
.ptype	The prototype of the data structure to return

Value

A lens that selects the specified elements

names_l *Names lens*

Description

Lens into the names attribute of an object.

Usage

```
names_l
```

Format

An object of class tinylens::lens (inherits from S7_object) of length 1.

over *Modify data through a lens*

Description

This function applies a lens to a data structure and modifies the focused part.

Usage

```
over(d, l, f)
```

Arguments

d	The data structure to modify
l	The lens to apply
f	The function to apply

Value

The modified data structure

over_map *Map a function over a list lens*

Description

Apply a function to each element of a list returned by a lens. Using over in such cases would require a "lifted" function, which is often unergonomic.

Usage

```
over_map(d, l, f)
```

Arguments

d	The data structure to modify
l	The list-returning lens to apply
f	The function to apply to each element of the list

Value

The modified data structure

rows_l *Rows lens*

Description

This function returns a lens that selects the specified rows.

Usage

```
rows_l(idx)
```

Arguments

idx	The rows to select

Value

A lens that selects the specified rows

select_l *include verbs.R include lens.R Select lens*

Description

This function returns a lens that selects the specified columns.

Usage

```
select_l(...)
```

Arguments

...	Columns to select

Value

A lens that selects the specified columns

set *Set data through a lens*

Description

This function applies a lens to a data structure and sets the focused part.

Usage

```
set(d, l, x)
```

Arguments

d	The data structure to set
l	The lens to apply
x	The value to set

Value

The modified data structure

slice_l *Slice lens*

Description

Lens into a slice of a vector.

Usage

```
slice_l(idx)
```

Arguments

idx	Indices of the elements to lens into

Details

This lens performs indexing using single bracket notation, i.e., `x[idx]`.

Value

A lens that selects the specified slice

vec_data_l *Vector data lens*

Description

Allows mutation of vector data while preserving attributes, e.g., labels or names.

Usage

```
vec_data_l
```

Format

An object of class `tinylens::lens` (inherits from `S7_object`) of length 1.

Examples

```
x <- c(a = "foo1", b = "bar2")
view(x, vec_data_l)
set(x, vec_data_l, c("foo2", "bar3"))
```

view *View data through a lens*

Description

This function applies a lens to a data structure and returns the focused part.

Usage

```
view(d, l)
```

Arguments

`d`	The data structure to view
`l`	The lens to apply

Value

The part of the data structure focused by the lens

where_il *Predicate ilens*

Description

Illegal lens into elements of a vector that satisfy a predicate.

Usage

```
where_il(p)
```

Arguments

p	A predicate function

Value

A lens that selects the elements that satisfy the predicate

%.% *Compose two lenses*

Description

The resulting lens first applies the *left* lens, then the right lens.

Usage

```
l %.% m
```

Arguments

l	First lens
m	Second lens

Value

A new lens

Index