

Package: leaflet.indoor (via r-universe)

October 19, 2026

Title Multi-Level Indoor Maps with 'Leaflet'

Version 0.2.0

Description Adds accessible floor controls and multi-level indoor vector layers to maps created with the 'leaflet' package. Simple features, GeoJSON, simple feature geometry collections, and optional 'SpatVector' objects can be displayed with one active floor per indoor data set. Room identifiers can also be associated with portable photo carousels whose images remain paired with plain-text captions. Optional room comments and attributed testimonials open from customizable, accessible map icons.

License MIT + file LICENSE

URL <https://aureliennicosiaulaval.github.io/leaflet-indoor/>,
<https://github.com/AurelienNicosiaULaval/leaflet-indoor>

BugReports <https://github.com/AurelienNicosiaULaval/leaflet-indoor/issues>

Depends R (>= 4.1.0)

Imports htmltools (>= 0.5.7), jsonlite (>= 1.8.8), leaflet (>= 2.2.0),
sf (>= 1.0-0)

Suggests covr, devtools, knitr, lintr, pkgdown, quarto, rmarkdown,
shiny, spelling, terra, testthat (>= 3.0.0), urlchecker, withr,
xml2

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Language en-US

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.1.0

LazyData true

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev make libpng-dev libuv1-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

Repository <https://aureliennicosiaulaval.r-universe.dev>
Date/Publication 2026-10-07 11:45:06 UTC
RemoteUrl <https://github.com/AurelienNicosiaULaval/leaflet-indoor>
RemoteRef HEAD
RemoteSha 7198508a63e15ea0caa2b0758b7e538f19d60053

Contents

addIndoor	2
addIndoorControl	4
clearIndoor	5
indoor_demo	5
indoor_demo_geo	6
indoorCommentCatalog	6
indoorCommentOptions	7
indoorControlOptions	8
indoorOptions	9
indoorPhotoCatalog	10
indoorPhotoOptions	11
removeIndoorControl	12
setIndoorLevel	12

Index	13
--------------	-----------

addIndoor	<i>Add multi-level indoor features</i>
-----------	--

Description

Adds points, lines, and polygons organized by floor to a `leaflet::leaflet()` map. Exactly one floor is visible for each `dataset_id`.

Usage

```
addIndoor(  
  map,  
  data = leaflet::getMapData(map),  
  level = "level",  
  dataset_id = "indoor",  
  level_order = NULL,  
  initial_level = NULL,  
  layerId = NULL,  
  label = NULL,  
  popup = NULL,  
  style = NULL,  
  options = indoorOptions(),
```

```

labelOptions = leaflet::labelOptions(),
popupOptions = leaflet::popupOptions(),
missing_level = c("error", "drop"),
crs = c("auto", "geographic", "simple"),
photos = NULL,
photoOptions = indoorPhotoOptions(),
comments = NULL,
commentOptions = indoorCommentOptions()
)

```

Arguments

map	A Leaflet map or proxy.
data	An sf, sfc, optional SpatVector, or GeoJSON object. The default uses the data supplied to <code>leaflet::leaflet()</code> or <code>leaflet::leafletProxy()</code> .
level	For sf and GeoJSON, the floor column or property name. For a bare sfc, an explicit vector or list of floor values.
dataset_id	Non-empty identifier scoped to this map.
level_order	Optional vector containing every observed floor exactly once, from lowest to highest.
initial_level	Initial floor. The default is "0" when present, then the first floor in level_order.
layerId	Optional feature identifiers. Formulas are evaluated against data.
label, popup	Optional tooltip and popup content. Plain strings are escaped; wrap trusted markup in <code>htmltools::HTML()</code> .
style	Optional named list of fixed, vector, or formula-driven Leaflet path options.
options	Options created by <code>indoorOptions()</code> .
labelOptions, popupOptions	Leaflet tooltip and popup options.
missing_level	Either "error" or "drop".
crs	Coordinate interpretation. "auto" detects the map CRS for ordinary maps. Use "simple" explicitly for local coordinates sent to a proxy.
photos	Optional photo catalogue created by <code>indoorPhotoCatalog()</code> , or a data frame with equivalent layerId, src, caption, and optional alt columns. Catalogue identifiers must match layerId values.
photoOptions	Carousel labels created by <code>indoorPhotoOptions()</code> .
comments	Optional catalogue created by <code>indoorCommentCatalog()</code> , or a data frame with equivalent columns. Identifiers must match layerId.
commentOptions	Icon and labels created by <code>indoorCommentOptions()</code> .

Value

The input map or proxy, with an indoor method queued.

Examples

```

leaflet::leaflet(
  indoor_demo,
  options = leaflet::leafletOptions(
    crs = leaflet::leafletCRS("L.CRS.Simple")
  )
) |>
leaflet::addMapPane("indoor", 420) |>
addIndoor(
  initial_level = "0",
  layerId = ~feature_id,
  label = ~name,
  popup = ~description,
  crs = "simple",
  options = indoorOptions(pane = "indoor")
) |>
addIndoorControl(position = "bottomright")

```

addIndoorControl	<i>Add an accessible floor control</i>
------------------	--

Description

Add an accessible floor control

Usage

```

addIndoorControl(
  map,
  dataset_id = "indoor",
  control_id = dataset_id,
  position = "topright",
  options = indoorControlOptions()
)

```

Arguments

map	A Leaflet map or proxy.
dataset_id	Indoor data set controlled by these buttons.
control_id	Unique control identifier scoped to this map.
position	One of "topleft", "topright", "bottomleft", or "bottomright".
options	Options created by indoorControlOptions() .

Value

The input map or proxy.

clearIndoor	<i>Clear an indoor data set</i>
-------------	---------------------------------

Description

Removes all features for one `dataset_id`. Linked controls remain present and show their configured empty state.

Usage

```
clearIndoor(map, dataset_id = "indoor")
```

Arguments

<code>map</code>	A Leaflet map or proxy.
<code>dataset_id</code>	Indoor data set to clear.

Value

The input map or proxy.

indoor_demo	<i>Synthetic multi-level indoor features in local coordinates</i>
-------------	---

Description

A synthetic, freely reusable indoor plan created for package examples and tests. It does not describe a real building. The mixed sf object contains points, lines, polygons, and multiple geometries. Its level list-column includes an elevator and stairs assigned to several floors.

Usage

```
indoor_demo
```

Format

An sf object with 10 features and 7 variables:

feature_id Stable synthetic feature identifier.

name Short feature label.

description Popup-ready description.

kind Feature category.

fill Suggested fill color.

level List-column of floor values.

geometry Local Cartesian geometry with an undefined CRS.

Source

Generated by data-raw/indoor_demo.R.

indoor_demo_geo	<i>Synthetic multi-level indoor features in geographic coordinates</i>
-----------------	--

Description

A geographic companion to [indoor_demo](#), constructed directly in EPSG:4326 for offline examples. It is synthetic and does not describe a real building.

Usage

```
indoor_demo_geo
```

Format

An sf object with the same attributes and geometry types as [indoor_demo](#), using EPSG:4326.

Source

Generated by data-raw/indoor_demo.R.

indoorCommentCatalog	<i>Create a catalogue of room comments or testimonials</i>
----------------------	--

Description

Each row associates a plain-text comment with a feature identifier supplied to [addIndoor\(\)](#). Repeating an identifier displays several comments in row order. Comments open from a separate map icon, leaving room popups and photo carousels available through the room itself.

Usage

```
indoorCommentCatalog(
  layerId,
  text,
  author = "",
  date = "",
  source = "",
  title = ""
)
```

Arguments

layerId	Feature identifiers matching layerId in <code>addIndoor()</code> .
text	Non-empty plain-text comments.
author, date	Optional plain-text attribution and date labels. An empty string omits the field. Dates are displayed as supplied.
source	Optional absolute HTTP(S) source URLs, or empty strings.
title	Optional plain-text headings, for example to distinguish a published excerpt from a summary.

Details

Fields of length one are recycled to the number of comments. Content is rendered as text, including strings wrapped in `HTML()`. Authors are responsible for permission and accurate attribution when displaying third-party material.

Value

A data frame of class `leaflet_indoor_comments`.

Examples

```
comments <- indoorCommentCatalog(  
  layerId = c("room-01", "room-01"),  
  text = c("A quiet place to read.", "The entrance is easy to find."),  
  author = c("Example visitor A", "Example visitor B")  
)  
comments
```

indoorCommentOptions	<i>Options for room comment icons</i>
----------------------	---------------------------------------

Description

Options for room comment icons

Usage

```
indoorCommentOptions(  
  icon = "comment",  
  color = "#ffffff",  
  background_color = "#7c3aed",  
  show = TRUE,  
  marker_label = "Read room comments",  
  popup_label = "Room comments",  
  source_label = "Read source",  
  size = 36,
```

```

    min_size = 12,
    fit_to_room = TRUE,
    placement = "edge"
)

```

Arguments

icon	One of "comment", "quote", or "info" for bundled vector icons, or a short plain-text symbol such as "\u2605" or "\u270e".
color, background_color	Icon foreground and background colors as hexadecimal #RGB or #RRGGBB strings.
show	Whether to display comment icons.
marker_label	Accessible action label for a comment icon. The room label, when supplied, is appended automatically.
popup_label	Heading displayed above the room comments.
source_label	Label for each comment's source link.
size	Maximum visual badge diameter in CSS pixels.
min_size	Minimum visual badge diameter when fitting to a room.
fit_to_room	Scale the badge and its icon with the room's projected size as the map zoom changes. Point features retain size. The clickable target remains at least 44 pixels for touch and keyboard access.
placement	"edge" places the icon just outside the room's exterior boundary, with a short connector, leaving the room available for clicks. "center" places it at the interior reference point. Point and line features without an exterior ring use an offset from their reference point.

Value

An object of class `leaflet_indoor_comment_options`.

Examples

```

indoorCommentOptions(icon = "quote", background_color = "#7c3aed")
indoorCommentOptions(icon = "\u2605", marker_label = "Lire les témoignages",
  popup_label = "Témoignages", source_label = "Source")

```

indoorControlOptions *Options for an indoor floor control*

Description

Options for an indoor floor control

Usage

```
indoorControlOptions(
  title = "Floor",
  descending = TRUE,
  max_height = 280,
  empty_label = "No levels"
)
```

Arguments

title	Accessible title displayed above the floor buttons.
descending	Display higher floors above lower floors.
max_height	Maximum button-list height in CSS pixels.
empty_label	Text displayed when the linked data set has no floors.

Value

An object of class `leaflet_indoor_control_options`.

indoorOptions	<i>Options for indoor layers</i>
---------------	----------------------------------

Description

Create the style and interaction options used by `addIndoor()`. The three geometry-specific lists use Leaflet path option names. Values supplied via the `style` argument of `addIndoor()` override these defaults.

Usage

```
indoorOptions(
  point = NULL,
  line = NULL,
  polygon = NULL,
  pane = NULL,
  interactive = TRUE
)
```

Arguments

point, line, polygon	Named lists of options for point, line, and polygon geometries.
pane	Optional Leaflet pane name.
interactive	Whether features receive mouse and keyboard events.

Value

An object of class `leaflet_indoor_options`.

<code>indoorPhotoCatalog</code>	<i>Create an indoor photo catalogue</i>
---------------------------------	---

Description

Creates the tabular association used by `addIndoor()` to attach one or more photos to indoor features. Each row associates a `layerId` with one image and its caption. Repeating a `layerId` creates a carousel in row order.

Usage

```
indoorPhotoCatalog(layerId, src, caption, alt = caption)
```

Arguments

<code>layerId</code>	Feature identifiers matching the values supplied to the <code>layerId</code> argument of <code>addIndoor()</code> . A scalar is recycled.
<code>src</code>	Image sources. These may be HTTP(S) URLs, relative URLs, image data URIs, or paths to existing PNG, JPEG, GIF, WebP, or SVG files.
<code>caption</code>	Plain-text captions displayed immediately below their images.
<code>alt</code>	Plain-text alternative text. The default uses <code>caption</code> .

Details

Existing local image files are embedded as data URIs so saved widgets, vignettes, Quarto documents, and Shiny applications do not depend on the original file path. Web URLs and relative URLs are retained unchanged.

Value

A data frame of class `leaflet_indoor_photos` with columns `layerId`, `src`, `caption`, and `alt`.

Examples

```
photos <- indoorPhotoCatalog(
  layerId = c("room-01", "room-01"),
  src = c(
    "https://example.org/room-01-a.jpg",
    "https://example.org/room-01-b.jpg"
  ),
  caption = c("Entrance view", "View from the windows")
)
photos
```

indoorPhotoOptions	<i>Options for indoor photo carousels</i>
--------------------	---

Description

Controls the visible and accessible labels used by photo carousels created when `addIndoor()` receives a non-empty photos catalogue. Labels can be translated without changing the catalogue.

Usage

```
indoorPhotoOptions(  
  previous_label = "Previous",  
  next_label = "Next",  
  enlarge_label = "Enlarge",  
  close_label = "Close",  
  carousel_label = "Room photos",  
  dialog_label = "Enlarged room photo",  
  counter_label = "Photo {current} of {total}",  
  unavailable_label = "Image unavailable"  
)
```

Arguments

<code>previous_label</code>	Label for the previous-photo button.
<code>next_label</code>	Label for the next-photo button.
<code>enlarge_label</code>	Label for the button that opens the current photo and caption in a larger dialog.
<code>close_label</code>	Label for the dialog close button.
<code>carousel_label</code>	Accessible label for the carousel.
<code>dialog_label</code>	Accessible label for the enlarged-photo dialog.
<code>counter_label</code>	Counter template containing {current} and {total}.
<code>unavailable_label</code>	Text shown when an image cannot be loaded.

Value

An object of class `leaflet_indoor_photo_options`.

removeIndoorControl	<i>Remove an indoor floor control</i>
---------------------	---------------------------------------

Description

Remove an indoor floor control

Usage

```
removeIndoorControl(map, control_id = "indoor")
```

Arguments

map	A Leaflet map or proxy.
control_id	Control to remove.

Value

The input map or proxy.

setIndoorLevel	<i>Set the active indoor floor</i>
----------------	------------------------------------

Description

Set the active indoor floor

Usage

```
setIndoorLevel(map, level, dataset_id = "indoor")
```

Arguments

map	A Leaflet map or proxy.
level	One non-missing floor value.
dataset_id	Indoor data set to update.

Value

The input map or proxy.

Index

* datasets

- indoor_demo, [5](#)
- indoor_demo_geo, [6](#)

- addIndoor, [2](#)
- addIndoor(), [6](#), [7](#), [9–11](#)
- addIndoorControl, [4](#)

- clearIndoor, [5](#)

- htmltools::HTML(), [3](#)

- indoor_demo, [5](#), [6](#)
- indoor_demo_geo, [6](#)
- indoorCommentCatalog, [6](#)
- indoorCommentCatalog(), [3](#)
- indoorCommentOptions, [7](#)
- indoorCommentOptions(), [3](#)
- indoorControlOptions, [8](#)
- indoorControlOptions(), [4](#)
- indoorOptions, [9](#)
- indoorOptions(), [3](#)
- indoorPhotoCatalog, [10](#)
- indoorPhotoCatalog(), [3](#)
- indoorPhotoOptions, [11](#)
- indoorPhotoOptions(), [3](#)

- leaflet::leaflet(), [2](#), [3](#)
- leaflet::leafletProxy(), [3](#)

- removeIndoorControl, [12](#)

- setIndoorLevel, [12](#)