

# Package ‘spatialCatalogueViewer’

February 25, 2026

**Type** Package

**Title** A 'Shiny' Tool to Create Interactive Catalogues for Geospatial Data

**Version** 0.2.0

**Date** 2026-02-25

**Author** Sebastien Plutniak [aut, cre] (ORCID:  
<<https://orcid.org/0000-0002-6674-3806>>)

**Maintainer** Sebastien Plutniak <[sebastien.plutniak@posteo.net](mailto:sebastien.plutniak@posteo.net)>

**Description** Seamlessly create interactive online catalogues for geospatial data. Items can be mapped as points or areas and retrieved using either a map or a dynamic table with search form and optional column filters.

**License** GPL-3

**Repository** CRAN

**Encoding** UTF-8

**Imports** leaflet, shiny, shinythemes, DT,

**Depends** R (>= 4.1.0)

**URL** <https://github.com/sebastien-plutniak/spatialCatalogueViewer>

**BugReports** <https://github.com/sebastien-plutniak/spatialCatalogueViewer/issues>

**NeedsCompilation** no

**Date/Publication** 2026-02-25 21:20:09 UTC

## Contents

spatialCatalogueViewer . . . . .	2
<b>Index</b>	<b>5</b>

---

 spatialCatalogueViewer

*spatialCatalogueViewer*


---

## Description

Launch the spatialCatalogueViewer application.

## Usage

```
spatialCatalogueViewer(data = NULL,
  text.title = NULL, text.left = NULL, text.top = NULL, text.bottom = NULL,
  map.provider = "Stadia.StamenTerrainBackground",
  map.set.lon = 0, map.set.lat = 0,
  map.legend.variable = NULL, map.legend.labels = NULL,
  map.legend.colors = NULL, map.height = 500,
  map.area.fill.color = "grey", map.area.fill.opacity = 0.5,
  map.show.areas = "never", map.min.zoom = NULL,
  table.hide.columns = NULL, table.filter = "none", table.pageLength = NULL,
  data.download.button = TRUE,
  tabs.contents = NULL,
  theme = "cosmo", css = NULL, js = NULL)
```

## Arguments

<code>data</code>	dataframe. The data. See details for mandatory columns.
<code>text.title</code>	character. HTML text to use for the title.
<code>text.left</code>	character. HTML text to use in the columns left to the map.
<code>text.top</code>	character. HTML text to use above the map.
<code>text.bottom</code>	character. HTML text to use below the map.
<code>map.provider</code>	character. Name of the map provider, see <a href="https://leaflet-extras.github.io/leaflet-providers/preview/">https://leaflet-extras.github.io/leaflet-providers/preview/</a> for possible values.
<code>map.set.lon</code>	integer. Initial longitude to center the map, a value in [-90, 90].
<code>map.set.lat</code>	integer. Initial latitude to center the map, a value in [-90, 90].
<code>map.legend.variable</code>	character. The name of the column to use for the legend. It also defines the legend title.
<code>map.legend.labels</code>	character. A vector with names for the legend's items. Must have the same length than 'map.legend.colors'.
<code>map.legend.colors</code>	character. A vector of color names for the legend's item. Must have the same length than 'map.legend.labels'. See <code>colors()</code> for a list of possible values.
<code>map.height</code>	integer. Height in px of the map.

<code>map.area.fill.color</code>	character. A color name to color the areas. See <code>colors()</code> for a list of possible values.
<code>map.area.fill.opacity</code>	numerical. Opacity of the areas, a value between 0 and 1.
<code>map.show.areas</code>	character or logical. One of TRUE, FALSE, 'always', 'never'. If TRUE or FALSE, a checkbox is displayed under the map.
<code>map.min.zoom</code>	integer. Minimal zoom for the map, between 1 and 12.
<code>table.hide.columns</code>	character. Name(s) of the column(s) to hide in the interactive table. Note that these columns can be searched using the 'Search' form anyways.
<code>table.filter</code>	character. Whether and where to use column filters: 'none': no filters, 'bottom' / 'top': put column filters at the bottom or top of the table.
<code>table.pageLength</code>	integer. Number of rows displayed on each page of the table.
<code>data.download.button</code>	logical. Display or not a button to download the dataset.
<code>tabs.contents</code>	character. A named list with the contents of the optional additional tabs. List elements' names are used as tab labels.
<code>theme</code>	character. Name of the shinytheme to use.
<code>css</code>	character. CSS code.
<code>js</code>	character. Javascript code.

## Details

This function launches the 'spatialCatalogueViewer' application.

The dataframe for data has mandatory and optional columns:

Mandatory:

- `resource.name`: Name of the resource (HTML contents is possible).
- `coordinates`: Either 'lon' and 'lat' columns or 4 columns named 'bbox.lon1', 'bbox.lat1', 'bbox.lon2', 'bbox.lat2' (where lon1 and lat1 refers to the north-west corner of the area and lon2 and lat2 to the south-east corner).

Optional:

- `resource.popup`: HTML contents for the map pop-up.

## Value

Launch the 'spatialCatalogueViewer' Shiny application.

## Author(s)

Sebastien Plutniak <sebastien.plutniak at posteo.net>

**See Also**

DT leaflet leaflet.extra shiny [shinytheme](#)

**Examples**

```
if(interactive()){  
df <- data.frame(lat = c(43.6033, 49.89441, 43.2974),  
                 lon = c(1.4444, 2.29649, 5.3703),  
                 resource.name = c("Toulouse", "Amiens", "Marseille"),  
                 population = c(511684, 135429, 877215),  
                 region = c("Occitanie", "Picardie", "Provence")  
)  
# launch the app:  
spatialCatalogueViewer(data = df,  
                        map.set.lon = 2.37, map.set.lat = 47.29, map.min.zoom = 5)  
}
```

# Index

shinytheme, [4](#)

spatialCatalogueViewer, [2](#)