

Package ‘webmap’

September 1, 2023

Title Create Interactive Web Maps Using 'The National Map' Services

Version 1.0.3

Description Creates interactive web maps using the 'JavaScript' 'Leaflet' library with base layers of 'The National Map' ('TNM'). 'TNM' services provide access to base geospatial information that describes the landscape of the United States and its territories. This package is dependent on, and intended to be used with, the 'leaflet' package.

Depends R (>= 4.1)

Imports checkmate, htmltools, htmlwidgets, leaflet

Suggests connectapi, covr, DT, knitr, pkgbuild, pkgdown, pkgload, rcmdcheck, remotes, roxygen2, sf, tinytest

License MIT + file LICENSE

URL <https://rconnect.usgs.gov/INLPO/webmap-main/>,
<https://code.usgs.gov/inl/webmap>

BugReports <https://code.usgs.gov/inl/webmap/-/issues>

Encoding UTF-8

VignetteBuilder knitr

BuildVignettes true

RoxygenNote 7.2.3

NeedsCompilation no

Author Jason C. Fisher [aut, cre] (<<https://orcid.org/0000-0001-9032-8912>>),
Stefano Cudini [ctb, cph] (leaflet-search library),
John Firebaugh [ctb, cph] (Leaflet.fullscreen library)

Maintainer Jason C. Fisher <jfisher@usgs.gov>

Repository CRAN

Date/Publication 2023-09-01 11:30:02 UTC

R topics documented:

add_cluster_button	2
add_fullscreen_button	3
add_home_button	4
add_legend	5
add_search_button	6
make_map	8

Index	12
--------------	-----------

add_cluster_button	<i>Add cluster control button to a web map</i>
--------------------	--

Description

Add a button to a [Leaflet](#) map to toggle marker clusters on and off.

Usage

```
add_cluster_button(map, cluster_id, position = "topleft")
```

Arguments

map	'leaflet'. Map widget object
cluster_id	'character' string. Identification for the marker cluster layer.
position	'character' string. Position of the button on the web map. Possible values are "topleft", "topright", "bottomleft", and "bottomright".

Value

A new HTML web map with added element, an object of class 'leaflet'.

Author(s)

J.C. Fisher, U.S. Geological Survey, Idaho Water Science Center

See Also

[make_map](#) function for creating a map widget.

Examples

```
# read city point locations from GeoJSON file
city <- system.file("extdata/city.geojson", package = "webmap") |>
  sf::st_read()

# create web map and add button to toggle marker clusters on and off
map <- make_map("Topo") |>
  leaflet::addMarkers(
    label = ~name,
    popup = ~name,
    clusterOptions = leaflet::markerClusterOptions(
      showCoverageOnHover = FALSE
    ),
    clusterId = "cluster",
    group = "marker",
    data = city
  ) |>
  add_cluster_button(cluster_id = "cluster")

# print web map
map
```

add_fullscreen_button *Add full-screen button to a web map*

Description

Add a button to a **Leaflet** map that toggles full screen on and off. Functionality provided by the **leaflet-fullscreen** plugin for Leaflet.

Usage

```
add_fullscreen_button(map, pseudo_fullscreen = FALSE, position = "topleft")
```

Arguments

map	'leaflet'. Map widget object
pseudo_fullscreen	'logical' flag. Whether to fullscreen to page width and height.
position	'character' string. Position of the button on the web map. Possible values are "topleft", "topright", "bottomleft", and "bottomright".

Value

A new HTML web map with added element, an object of class 'leaflet'.

Author(s)

J.C. Fisher, U.S. Geological Survey, Idaho Water Science Center

See Also

[make_map](#) function for creating a map widget.

Examples

```
make_map("Topo") |> add_fullscreen_button()
```

add_home_button	<i>Add home button to a web map</i>
-----------------	-------------------------------------

Description

Add a button to a [Leaflet](#) map that zooms to the provided map extent.

Usage

```
add_home_button(map, extent = NULL, position = "topleft")
```

Arguments

map	'leaflet'. Map widget object
extent	'bbox', or 'numeric' vector of length four, with xmin, xmax, ymin and ymax values. Extent object representing a rectangular geographical area on the map. The extent must be specified in the coordinate reference system (CRS) of the web map, usually in latitude and longitude using WGS 84 (also known as EPSG:4326). By default, the extent will be automatically determined from latitudes and longitudes of the map elements.
position	'character' string. Position of the button on the web map. Possible values are "topleft", "topright", "bottomleft", and "bottomright".

Value

A new HTML web map with added element, an object of class 'leaflet'.

Author(s)

J.C. Fisher, U.S. Geological Survey, Idaho Water Science Center

See Also

[make_map](#) function for creating a map widget.

Examples

```
make_map("Topo") |>
  add_home_button(
    extent = c(-124.409591, -114.131211, 32.534156, 42.009518) # California
  )
```

`add_legend`*Add legend to a web map*

Description

Add a legend to a [Leaflet](#) map.

Usage

```
add_legend(  
  map,  
  labels,  
  colors,  
  radius,  
  opacity = 0.5,  
  symbol = c("square", "circle"),  
  title = "EXPLANATION",  
  position = "topright"  
)
```

Arguments

<code>map</code>	' leaflet '. Map widget object
<code>labels</code>	'character' vector. Labels in the legend.
<code>colors</code>	'character' vector. HTML colors corresponding to labels.
<code>radius</code>	'numeric' number. Border radius of symbols in the legend, in pixels.
<code>opacity</code>	'numeric' number. Opacity of symbols in the legend, from 0 to 1.
<code>symbol</code>	'character' string. Symbol type in the legend, either "square" or "circle".
<code>title</code>	'character' string. Legend title
<code>position</code>	'character' string. Position of the button on the web map. Possible values are "topleft", "topright", "bottomleft", and "bottomright".

Value

A new HTML web map with added element, an object of class 'leaflet'.

Author(s)

J.C. Fisher, U.S. Geological Survey, Idaho Water Science Center

See Also

[make_map](#) function for creating a map widget.

Examples

```
# read city point locations from GeoJSON file
city <- system.file("extdata/city.geojson", package = "webmap") |>
  sf::st_read()

# define marker colors based on whether a city serves as a capital
color <- c(
  "Non-capital" = "green",
  "Capital" = "red"
)

# print web map with city circle markers and legend
make_map("Topo") |>
  leaflet::addCircleMarkers(
    radius = 6,
    color = "white",
    weight = 1,
    opacity = 1,
    fillColor = as.character(color[(city$capital > 0) + 1]),
    fillOpacity = 1,
    fill = TRUE,
    data = city
  ) |>
  add_legend(
    labels = names(color),
    colors = color,
    radius = 5,
    opacity = 1,
    symbol = "circle"
  )
```

add_search_button *Add search button to a web map*

Description

Add a button to a **Leaflet** map to search markers/features location by property. Functionality provided by the **leaflet-search** plugin for Leaflet.

Usage

```
add_search_button(
  map,
  group,
  property_name = "label",
  zoom = NULL,
  text_placeholder = "Search...",
  open_popup = FALSE,
  position = "topleft"
)
```

Arguments

map	'leaflet'. Map widget object
group	'character' string. Name of the group whose features will be searched.
property_name	'character' string. Property name used to describe markers, such as, "label" and "popup".
zoom	'integer' count. Zoom level for move to location after marker found in search.
text_placeholder	'character' string. Message to show in search element.
open_popup	'logical' flag. Whether to open the marker popup associated with the searched for marker.
position	'character' string. Position of the button on the web map. Possible values are "topleft", "topright", "bottomleft", and "bottomright".

Value

A new HTML web map with added element, an object of class 'leaflet'.

Author(s)

J.C. Fisher, U.S. Geological Survey, Idaho Water Science Center

See Also

[make_map](#) function for creating a map widget.

Examples

```
# read city point locations from GeoJSON file
city <- system.file("extdata/city.geojson", package = "webmap") |>
  sf::st_read()

# create web map and add button to search city names
map <- make_map("Topo") |>
  leaflet::addMarkers(
    label = ~name,
    popup = ~name,
    group = "marker",
    data = city
  ) |>
  add_search_button(
    group = "marker",
    zoom = 15,
    text_placeholder = "Search city names..."
  )

# print web map
map
```

 make_map

Create a web map using TNM services

Description

Create a **Leaflet** map widget with base maps offered through The National Map (**TNM**). Information about the content of these base maps can be found within the **TNM Base Maps** document. The map widget can be rendered on HTML pages generated from R Markdown, Shiny, or other applications.

Usage

```
make_map(maps, ..., protocol = c("WMTS", "WMS"), collapse = FALSE)
```

Arguments

maps	'character' vector. TNM base maps to include in the web map. Choices include "Topo", "Imagery Only", "Imagery Topo", "Hydrography", "Shaded Relief", and "Blank". See 'Details' section for a description of each base map. By default, all base maps are included. The one exception is the "Blank" map, which is only accessible using a Web Map Service (WMS), see protocol argument.
...	Arguments to be passed to the <code>leaflet::leaflet</code> function.
protocol	'character' string. Standard protocol for serving pre-rendered georeferenced TNM map tiles. Select "WMTS" for the Web Map Tile Service (the default) and "WMS" for the Web Map Service.
collapse	'logical' flag. Whether the layers control should be rendered as an icon that expands when hovered over. Default is FALSE.

Details

Map **service endpoints** are offered through TNM with no use restrictions. However, map content is limited to the United States and territories. This function integrates TNM endpoint services within an interactive web map using **Leaflet for R**.

TNM base maps include:

- Topo combines the most current TNM data, and other public-domain data, into a multi-scale topographic reference map. Data includes boundaries, geographic names, transportation, con-



tours, hydrography, land cover, shaded relief, and bathymetry.

- Imagery Only is the orthoimagery in TNM. Orthoimagery data typically are high resolution aerial images that combine the visual attributes of an aerial photograph with the spatial accuracy and reliability of a planimetric map. USGS digital orthoimage resolution may vary from



6 inches to 1 meter.

- Imagery Topo is the orthoimagery in TNM as a backdrop, and a limited selection of topo-



graphic data (boundaries, names, transportation, contours, and hydrography).

- Hydrography is a cartographic representation of the **National Hydrography Dataset (NHD)**. The NHD is a comprehensive set of digital geospatial data that encodes information about naturally occurring and constructed bodies of surface water, paths through which water flows, re-



lated features such as stream gages and dams, and additional hydrologic information.

- Shaded Relief is a terrain representation in the form of hillshades created from the **3D Elevation Program** (3DEP). 3DEP maintains a seamless dataset of best available raster elevation data, in the form of digital elevation models (DEMs) for the conterminous United States,



Alaska, Hawaii, and Territorial Islands of the United States.



- Blank consists of ocean tints to give the outline of land cover as an empty base map.

Value

An object of class 'leaflet', a hypertext markup language (HTML) map widget. See example for instructions on how to add additional graphic layers (such as points, lines, and polygons) to the map

widget. Graphic layers added to the web map must be in latitude and longitude using WGS 84 (also known as **EPSG:4326**).

Author(s)

J.C. Fisher, U.S. Geological Survey, Idaho Water Science Center

Examples

```
# define arbitrary coordinate locations in decimal degrees
pts <- rbind(
  c(-112.049705, 43.517810),
  c(-122.171257, 37.456526),
  c(-77.367458, 38.947206),
  c(-149.803565, 61.187905),
  c(-80.248344, 26.080860)
)

# create map widget and add markers at coordinate locations
map <- make_map() |>
  leaflet::addMarkers(pts[, 1], pts[, 2])

# print map widget
map

# print map of satellite imagery with a rectangle in the vicinity of UCLA
make_map(c("Imagery Only", "Topo"), collapse = TRUE) |>
  leaflet::addRectangles(
    lng1 = -118.456554,
    lat1 = 34.078039,
    lng2 = -118.436383,
    lat2 = 34.062717,
    fillColor = "transparent"
  )
```

Index

`add_cluster_button`, [2](#)
`add_fullscreen_button`, [3](#)
`add_home_button`, [4](#)
`add_legend`, [5](#)
`add_search_button`, [6](#)

`leaflet`, [2–5](#), [7](#)
`leaflet::leaflet`, [8](#)

`make_map`, [2](#), [4](#), [5](#), [7](#), [8](#)