Package ‘spatialwidget’

January 18, 2019

Type Package
Title Converts Spatial Data to Javascript Object Notation (JSON) for Use in Htmlwidgets
Version 0.2
Date 2019-01-18

Description Many packages use ‘htmlwidgets’ <https://CRAN.R-project.org/package=htmlwidgets> for interactive plotting of spatial data. This package provides functions for converting R objects, such as simple features, into structures suitable for use in ‘htmlwidgets’ mapping libraries.

License GPL-3
Depends R (>= 3.3.0)
Encoding UTF-8
LazyData true

Imports Rcpp
LinkingTo BH, colourvalues (>= 0.2.2), geojsonsf (>= 1.3.0), jsonify (>= 0.2.0), rapidjsonr, Rcpp
RoxygenNote 6.1.1
Suggests colourvalues, covr, geojsonsf, jsonify, jsonlite, knitr, rmarkdown, sf, testthat

VignetteBuilder knitr
NeedsCompilation yes

Author David Cooley [aut, cre]
Maintainer David Cooley <dcooley@symbolix.com.au>

Repository CRAN
Date/Publication 2019-01-18 09:40:03 UTC
**R topics documented:**

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>widget_arcs</td>
<td>2</td>
</tr>
<tr>
<td>widget_capitals</td>
<td>3</td>
</tr>
<tr>
<td>widget_line</td>
<td>3</td>
</tr>
<tr>
<td>widget_melbourne</td>
<td>4</td>
</tr>
<tr>
<td>widget_od</td>
<td>5</td>
</tr>
<tr>
<td>widget_point</td>
<td>5</td>
</tr>
<tr>
<td>widget_polygon</td>
<td>6</td>
</tr>
<tr>
<td>widget_roads</td>
<td>7</td>
</tr>
</tbody>
</table>

**Index**

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>widget_arcs</td>
<td>8</td>
</tr>
</tbody>
</table>

**Description**

A simple feature sf object with two sfc columns, "origin" and "destination"

**Usage**

`widget_arcs`

**Format**

A sf object with 199 observations and 6 variables

- **country_from** origin country
- **capital_from** origin capital
- **country_to** destination country
- **capital_to** destination capital
- **origin** sfc geometry column
- **destination** sfc geometry column
widget_capitals

Capital cities for each country

Description
A simple feature sf object containing the coordinates of 200 capital cities in the world

Usage

widget_capitals

Format
A sf object with 200 observations and 4 variables

country  country name
capital  capital name
gometry  sfc geometry column

widget_line

Widget Line

Description
Converts an ‘sf’ object with LINESTRING geometries into JSON for plotting in an htmlwidget

Usage

widget_line(data, stroke_colour = NULL, stroke_opacity = NULL,
           stroke_width = NULL, legend = TRUE, json_legend = TRUE)

Arguments
data  sf object
stroke_colour  string specifying column of sf to use for the stroke colour, or a single value to apply to all rows of data
stroke_opacity  string specifying column of sf to use for the stroke opacity, or a single value to apply to all rows of data
stroke_width  string specifying column of sf to use for the stroke width, or a single value to apply to all rows of data
legend  logical indicating if legend data will be returned
json_legend  logical indicating if the legend will be returned as json
Examples

```r
## use default stroke options
l <- widget_line( widget_roads, legend = TRUE )
```

### Description

A simple feature sf object of Polygons for Melbourne and the surrounding area

### Usage

`widget_melbourne`

### Format

A data frame with 397 observations and 7 variables

- **SA2_NAME**: statistical area 2 name of the polygon
- **SA3_NAME**: statistical area 3 name of the polygon
- **AREASQKM**: area of the SA2 polygon
- **geometry**: sfc geometry column

### Details


The data is released under a Creative Commons Attribution 2.5 Australia licence [https://creativecommons.org/licenses/by/2.5/au/](https://creativecommons.org/licenses/by/2.5/au/)

The data has been down-cast from MULTIPOLYGONS to POLYGONS.
widget_od

**Description**

Converts an ‘sf’ object with two POINT geometries into JSON for plotting in an htmlwidget

**Usage**

```r
widget_od(data, origin, destination, fill_colour = NULL,
fill_opacity = NULL, legend = TRUE, json_legend = TRUE)
```

**Arguments**

- `data` *sf object*
- `origin` string specifying the column of data containing the origin geometry
- `destination` string specifying the column of data containing the destination geometry
- `fill_colour` string specifying column of sf to use for the fill colour, or a single value to apply to all rows of data
- `fill_opacity` string specifying column of sf to use for the fill opacity, or a single value to apply to all rows of data
- `legend` logical indicating if legend data will be returned
- `json_legend` logical indicating if the legend will be returned as json

**Examples**

```r
l <- widget_od(data = widget_arcs, origin = "origin",
    destination = "destination", legend = FALSE)
```

widget_point

**Description**

Converts an ‘sf’ object with POINT geometries into JSON for plotting in an htmlwidget

**Usage**

```r
widget_point(data, fill_colour = NULL, fill_opacity = NULL,
    lon = NULL, lat = NULL, legend = TRUE, json_legend = TRUE)
```
Arguments

- **data**: sf object
- **fill_colour**: string specifying column of sf to use for the fill colour, or a single value to apply to all rows of data
- **fill_opacity**: string specifying column of sf to use for the fill opacity, or a single value to apply to all rows of data
- **lon**: string specifying the column of data containing the longitude. Ignored if using an sf object
- **lat**: string specifying the column of data containing the latitude. Ignored if using an sf object
- **legend**: logical indicating if legend data will be returned
- **json_legend**: logical indicating if the legend will be returned as json

Examples

```r
l <- widget_point( data = widget_capitals, legend = FALSE )
```

---

**Description**

Converts an ‘sf’ object with POLYGON geometries into JSON for plotting in an htmlwidget

**Usage**

```r
widget_polygon(data, stroke_colour = NULL, stroke_opacity = NULL,
               stroke_width = NULL, fill_colour = NULL, fill_opacity = NULL,
               legend = TRUE, json_legend = TRUE)
```

**Arguments**

- **data**: sf object
- **stroke_colour**: string specifying column of sf to use for the stroke colour, or a single value to apply to all rows of data
- **stroke_opacity**: string specifying column of sf to use for the stroke opacity, or a single value to apply to all rows of data
- **stroke_width**: string specifying column of sf to use for the stroke width, or a single value to apply to all rows of data
- **fill_colour**: string specifying column of sf to use for the fill colour, or a single value to apply to all rows of data
### widget_roads

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fill_opacity</td>
<td>string specifying column of sf to use for the fill opacity, or a single value to apply to all rows of data</td>
</tr>
<tr>
<td>legend</td>
<td>logical indicating if legend data will be returned</td>
</tr>
<tr>
<td>json_legend</td>
<td>logical indicating if the legend will be returned as json</td>
</tr>
</tbody>
</table>

#### Examples

```r
l <- widget_polygon( widget_melbourne, legend = FALSE )
l <- widget_polygon( widget_melbourne, fill_colour = "AREASQKm16", legend = TRUE )
```

---

**Description**

A simple feature sf object of roads in central Melbourne

**Usage**

```r
widget_roads
```

**Format**

An sf and data frame object with 18286 observations and 16 variables

**Details**

Obtained from [www.data.gov.au](http://www.data.gov.au) and distributed under the Creative Commons 4 License [https://creativecommons.org/licenses/by/4.0/](https://creativecommons.org/licenses/by/4.0/)
Index

*Topic datasets
  widget_arcs, 2
  widget_capitals, 3
  widget_melbourne, 4
  widget_roads, 7

widget_arcs, 2
widget_capitals, 3
widget_line, 3
widget_melbourne, 4
widget_od, 5
widget_point, 5
widget_polygon, 6
widget_roads, 7