Package 'leaflet.esri'

April 24, 2018

Type  Package
Title  'ESRI' Bindings for the 'leaflet' Package
Version  1.0.0
Description  An add-on package to the 'leaflet' package, which provides bindings for 'ESRI' services. This package allows a user to add 'ESRI' provided services such as 'MapService', 'ImageMapService', 'TiledMapService' etc. to a 'leaflet' map.
License  GPL-3 | file LICENSE
Depends  R (>= 3.1.0), leaflet (>= 2.0.0), leaflet.extras (>= 1.0.0)
Encoding  UTF-8
LazyData  true
Imports  htmltools
Suggests  htmlwidgets
URL  https://github.com/bhaskarvk/leaflet.esri
     https://bhaskarvk.github.io/leaflet.esri/
BugReports  https://github.com/bhaskarvk/leaflet.esri/issues
RoxygenNote  6.0.1
NeedsCompilation  no
Author  Bhaskar Karambelkar [aut, cre],
        Barret Schloerke [aut],
        Patrick Arlt [ctb, cph] (esri-leaflet, esri-leaflet-cluster, esri-leaflet-geocoder, and esri-leaflet-heatmap libraries),
        Rachel Nehmer [ctb, cph] (esri-leaflet-renderers library),
        Vladimir Agafonkin [ctb, cph] (leaflet.heat and simpleheat libraries),
        Leaflet [ctb, cph] (leaflet.markercluster library)
Maintainer  Bhaskar Karambelkar <bhaskarvk@gmail.com>
Repository  CRAN
Date/Publication  2018-04-23 22:31:56 UTC
R topics documented:

- `addEsriBasemapLayer` .......................................................... 2
- `addEsriDependency` .............................................................. 3
- `addEsriDynamicMapLayer` ...................................................... 3
- `addEsriFeatureLayer` ............................................................ 4
- `addEsriHeatmapFeatureLayer` .............................................. 6
- `addEsriImageMapLayer` .......................................................... 7
- `addEsriTiledMapLayer` ........................................................... 8
- `dynamicMapLayerOptions` ..................................................... 9
- `esriBasemapLabels` ............................................................. 10
- `esriBasemapLayers` ............................................................. 10
- `featureLayerOptions` ........................................................... 11
- `imageMapLayerOptions` ....................................................... 12
- `leaflet.esri` ................................................................. 13
- `tiledMapLayerOptions` ........................................................ 13

**:Index**: 14

---

### addEsriBasemapLayer

*Adds a ArcGIS Basemap layer*

**Description**

Adds a ArcGIS Basemap layer

**Usage**

```r
addEsriBasemapLayer(map, key, autoLabels = FALSE, layerId = NULL,
                    group = NULL, options = NULL)
```

**Arguments**

- `map` The leaflet map
- `key` ID of the layer
- `autoLabels` whether to show corresponding labels layer
- `layerId` Unique ID for the layer
- `group` The group this layer belongs to.
- `options` Basemap Layer Options. You can pass `tileOptions()`.

**Examples**

```r
leaflet() %>%
  addEsriBasemapLayer(esriBasemapLayers$Oceans, autoLabels = TRUE)
```
addEsriDependency

**Description**

Adds esri-leaflet dependency to the leaflet widget

- Adds esri-leaflet-clustered-feature-layer dependency to the leaflet widget
- Adds esri-leaflet-geocoder dependency to the leaflet widget
- Adds esri-leaflet-heatmap dependency to the leaflet widget
- Adds esri-leaflet-renderers dependency to the leaflet widget

**Usage**

```javascript
addEsriDependency(map)
addEsriClusterDependency(map)
addEsriGeocoderDependency(map)
addEsriHeatmapDependency(map)
addEsriRendererDependency(map)
```

**Arguments**

- **map**
  - The leaflet map widget

addEsriDynamicMapLayer

*Render and visualize Map Services from ArcGIS Online and ArcGIS Server.*

**Description**

Map Services are a way to expose the contents of a map as a web service and expose capabilities for exporting tile images, querying and identifying features and more. Also supports custom popups and identification of features.

**Usage**

```javascript
addEsriDynamicMapLayer(map, url, options = dynamicMapLayerOptions(),
                        popupFunction = NULL, popupOptions = NULL, layerId = NULL,
                        group = NULL)
```
addEsriFeatureLayer

Arguments

- **map**
  - The leaflet map.
- **url**
  - URL of the Map Service.
- **options**
  - options for the dynamic map layer.
- **popupFunction**
  - Uses the provided function to create a popup that will identify features whenever the map is clicked. Your function will be passed a GeoJSON FeatureCollection of the features at the clicked location and should return the appropriate HTML. If you do not want to open the popup when there are no results, return false.
- **popupOptions**
  - See **popupOptions**.
- **layerId**
  - A unique ID for the layer.
- **group**
  - The name of the group this layer should be added to.

Examples

```r
popupFunc <- htmlwidgets::JS(
  "function (error, featureCollection) {
    if (error || featureCollection.features.length === 0) {
      return false;
    } else {
      return "Risk Level: \" + featureCollection.features[0].properties.CLASS_DESC;
    }
  }
)

leaflet() %>% setView(-96.8, 38.5, 4) %>%
  addEsriBasemapLayer(esriBasemapLayers$Gray, autoLabels = TRUE) %>%
  addEsriDynamicMapLayer(
    url = paste0("https://maps7.arcgisonline.com/arcgis/rest/services/",
                
                "USDA_USFS_2014_Wildfire_Hazard_Potential/MapServer"),
    popupFunction = popupFunc)

## for more examples see
# browseURL(system.file("examples/dynamicMapLayers.R", package = "leaflet.esri"))
```

Description

FeatureLayer is used to visualize, style, query and edit vector geographic data hosted in both ArcGIS Online and published using ArcGIS Server. Copyright text from the service is added to map attribution automatically.
Usage

```javascript
addEsriFeatureLayer(map, url, useServiceSymbology = FALSE,
    options = featureLayerOptions(), layerId = NULL, group = NULL,
    markerType = NULL, markerIcons = NULL, markerIconProperty = NULL,
    markerOptions = leaflet::markerOptions(), clusterOptions = NULL,
    clusterId = NULL, labelProperty = NULL,
    labelOptions = leaflet::labelOptions(), popupProperty = NULL,
    popupOptions = leaflet::popupOptions(), stroke = TRUE, color = "#03F",
    weight = 5, opacity = 0.5, fill = TRUE, fillColor = color,
    fillOpacity = 0.2, dashArray = NULL, smoothFactor = 1, noClip = FALSE,
    pathOptions = leaflet::pathOptions(), highlightOptions = NULL,
    fitBounds = FALSE)
```

Arguments

- **map** The leaflet map
- **url** url of the FeatureService or MapService.
- **useServiceSymbology** whether to use the symbology set when a service was published.
- **options** options for the featurelayer
- **layerId** A unique ID for the layer.
- **group** The name of the group this layer should be added to. the same parameter under `addTiles`)
- **markerType** The type of marker. either "marker" or "circleMarker"
- **markerIcons** Icons for Marker.
- **markerIconProperty** The property of the feature to use for marker icon. Can be a JS function which accepts a feature and returns an index of markerIcons. In either case the result must be one of the indexes of markerIcons.
- **markerOptions** The options for markers Can be a single marker using `makeIcon` or a list of markers using `iconList`
- **clusterOptions** if not NULL, markers will be clustered using `Leaflet.markercluster`; you can use `markerClusterOptions()` to specify marker cluster options
- **clusterId** the id for the marker cluster layer
- **labelProperty** The property to use for the label. You can also pass in a JS function that takes in a feature and returns a text/HTML content.
- **labelOptions** A Vector of `labelOptions` to provide label options for each label. Default NULL
- **popupProperty** The property to use for popup content You can also pass in a JS function that takes in a feature and returns a text/HTML content.
- **popupOptions** A Vector of `popupOptions` to provide popups
- **stroke** whether to draw stroke along the path (e.g. the borders of polygons or circles)
- **color** stroke color
- **weight** stroke width in pixels
opacity stroke opacity (or layer opacity for tile layers)
fill whether to fill the path with color (e.g. filling on polygons or circles)
fillColor fill color
fillOpacity fill opacity
dashArray a string that defines the stroke dash pattern
smoothFactor how much to simplify the polyline on each zoom level (more means better performance and less accurate representation)
noclip whether to disable polyline clipping
pathOptions Options for shapes
highlightOptions Options for highlighting the shape on mouse over. you can use highlightOptions() to specify highlight
fitBounds Whether to set the maps bounds to fit the data in the featureLayer options

Examples

```r
gofer() %>%
  addEsriBasemapLayer(esriBasemapLayers$Streets) %>%
  setView(-122.667, 45.526, 13) %>%
  addEsriFeatureLayer(
    url = paste0("https://services.arcgis.com/r0o16HdIMeOBI4Mb/arcgis/rest/services/",
                  "Heritage_Trees_Portland/FeatureServer/0"),
    useServiceSymbology = TRUE,
    labelProperty = "COMMON_NAM", labelOptions = labelOptions(textsize = "12px"),
    popupProperty = JS(paste0("function(feature) {",
                              " return L.Util.template(",
                              " \"<h3>{COMMON_NAM}</h3><hr />\",
                              " <p>This tree is located at {ADDRESS} and its scientific name is {SCIENTIFIC}.</p>",
                              " \"",
                              " feature.properties",
                              " });
                              "})
                              "))
```
addEsriImageMapLayer

Usage

```javascript
addEsriHeatmapFeatureLayer(map, url, radius = 25, gradient = NULL,
    options = featureLayerOptions(), layerId = NULL, group = NULL)
```

Arguments

- **map**: The leaflet map
- **url**: URL of the FeatureService or MapService.
- **radius**: Radius for the heatmap
- **gradient**: The gradient
- **options**: options for the featureLayer
- **layerId**: A unique ID for the layer.
- **group**: The name of the group this layer should be added to. (the same parameter under `addTiles()`)

addEsriImageMapLayer

Render and visualize Image Services from ArcGIS Online and ArcGIS Server.

Description

Image Services provide access to raster data through a web service.

Usage

```javascript
addEsriImageMapLayer(map, url, options = imageMapLayerOptions(),
    popupFunction = NULL, popupOptions = NULL, layerId = NULL,
    group = NULL)
```

Arguments

- **map**: The leaflet map
- **url**: URL of the Image Service
- **options**: options for the image map layer.
- **popupFunction**: Uses the provided function to create a popup that will identify features whenever the map is clicked. Your function will be passed a GeoJSON FeatureCollection of the features at the clicked location and should return the appropriate HTML. If you do not want to open the popup when there are no results, return false.
- **popupOptions**: See `popupOptions`.
- **layerId**: A unique ID for the layer.
- **group**: The name of the group this layer should be added to.
addEsriTiledMapLayer

Examples

```r
leaflet() %>%
  addEsriBasemapLayer(esriBasemapLayers$Imagery) %>%
  setView(c(-120.23, 43.5, 5)) %>%
  addEsriImageMapLayer(
    url = paste0(
      "http://imagery.oregonexplorer.info/arcgis/rest/services/",
      "NAIP_2011/NAIP_2011_Dynamic/ImageServer"
    ),
    options = imageMapLayerOptions(bandIds = c(3, 0, 1))
  )
```

## for more examples see
# browseURL(system.file("examples/imageMapLayers.R", package = "leaflet.esri"))

---

**addEsriTiledMapLayer**  
*Access tiles from ArcGIS Online and ArcGIS Server to visualize and identify features.*

Description

If you have published a Feature Service in ArcGIS Online, it can be used to create a static set of tiles as well. You can find details about that process in the ArcGIS Online Help. Your map service must be published using the Web Mercator Auxiliary Sphere tiling scheme (WKID 102100/3857) and the default scale options used by Google Maps, Bing Maps and ArcGIS Online. Esri Leaflet will not support any other spatial reference for tile layers.

Usage

```r
addEsriTiledMapLayer(map, url, options = tiledMapLayerOptions(),
  layerId = NULL, group = NULL)
```

Arguments

- **map**  
The leaflet map
- **url**  
URL of the Map Service with a tile cache.
- **options**  
options for the tiledmap layer.
- **layerId**  
A unique ID for the layer.
- **group**  
The name of the group this layer should be added to.

Examples

```r
leaflet() %>%
  setView(c(-81.47, 30.70, 12)) %>%
  addEsriTiledMapLayer(
    url = "https://services.arcgisonline.com/ArcGIS/rest/services/USA_Topo_Maps/MapServer"
  )
```

## for more examples see
# browseURL(system.file("examples/tiledMapLayers.R", package = "leaflet.esri"))
**dynamicMapLayerOptions**

*Options for dynamic map layer.*

**Description**
Options for dynamic map layer.

**Usage**

```javascript
dynamicMapLayerOptions(format = "png24", transparent = TRUE, f = "json", attribution = ", layers = NULL, layerDefs = NULL, opacity = 1, position = "front", maxZoom = NULL, minZoom = NULL, dynamicLayers = NULL, token = NULL, proxy = NULL, useCors = TRUE, ...)
```

**Arguments**

- `format`: Output format of the image.
- `transparent`: Allow the server to produce transparent images.
- `f`: Server response content type.
- `attribution`: Attribution from service metadata copyright text is automatically displayed in Leaflet’s default control. This property can be used for customization.
- `layers`: An array of Layer IDs like [3, 4, 5] to show from the service.
- `layerDefs`: A string representing a query to run against the service before the image is rendered. This can be a string like "3:STATE_NAME="Kansas"" or an object mapping different queries to specific layers 3:"STATE_NAME="Kansas"", 2:"POP2007>25000".
- `opacity`: Opacity of the layer. Should be a value between 0 (completely transparent) and 1 (completely opaque).
- `position`: Position of the layer relative to other overlays.
- `maxZoom`: Closest zoom level the layer will be displayed on the map.
- `minZoom`: Furthest zoom level the layer will be displayed on the map.
- `dynamicLayers`: JSON object literal used to manipulate the layer symbology defined in the service itself. Requires a 10.1 (or above) map service which supports dynamicLayers requests.
- `token`: If you pass a token in your options it will be included in all requests to the service.
- `proxy`: URL of an ArcGIS API for JavaScript proxy or ArcGIS Resource Proxy to use for proxying requests.
- `useCors`: If this service should use CORS when making GET requests.
- `...`: extra options
esriBasemapLabels  
*Esri basemap labels.*

**Description**

Taken from https://esri.github.io/esri-leaflet/api-reference/layers/basemap-layer.html#optional-labels.

**Usage**

esriBasemapLabels

**Format**

An object of class list of length 7.

esriBasemapLayers  
*Esri Basemap Layers.*

**Description**

BasemapLayer is used to display Esri hosted basemaps and attributes data providers appropriately. The Terms of Use for Esri hosted services apply to all Leaflet applications. Taken from https://esri.github.io/esri-leaflet/api-reference/layers/basemap-layer.html.

**Usage**

esriBasemapLayers

**Format**

An object of class list of length 9.
featureLayerOptions | Options for featureLayers.

**Description**

Options for featureLayers.

**Usage**

```javascript
featureLayerOptions(where = NULL, minZoom = NULL, maxZoom = NULL,
cacheLayers = NULL, fields = NULL, from = NULL, to = NULL,
timeField = NULL, timeFilterMode = NULL, simplifyFactor = NULL,
precision = NULL, token = NULL, proxy = NULL, useCors = NULL,
renderer = NULL, ...)
```

**Arguments**

- **minZoom** Integer Minimum zoom level of the map that features will display. example: minZoom:0
- **maxZoom** Integer Maximum zoom level of the map that features will example: maxZoom:19
- **cacheLayers** Boolean Will remove layers from the internal cache when they are removed from the map.
- **fields** Array An array of fieldnames to pull from the service. Includes all fields by default. You should always specify the name of the unique id for the service. Usually either "FID" or "OBJECTID".
- **from** Date When paired with to defines the time range of features to display. Requires the Feature Layer to be time enabled.
- **to** Date When paired with from defines the time range of features to display. Requires the Feature Layer to be time enabled.
- **timeField** false The name of the field to lookup the time of the feature. Can be an object like start:"startTime", end:"endTime" or a string like "created".
- **timeFilterMode** "server" (default) or "client" Determines where features are filtered by time. By default features will be filtered by the server. If set to "client" all features are requested and filtered by the app before display.
- **simplifyFactor** Integer How much to simplify polygons and polylines. More means better performance, and less means more accurate representation.
**imageMapLayerOptions**

**token**  
String If you pass a token in your options it will be included in all requests to the service.

**proxy**  
URL of an ArcGIS API for JavaScript proxy or ArcGIS Resource Proxy to use for proxying requests.

**useCors**  
Boolean If this service should use CORS when making GET requests.

**renderer**  
L.svg or L.canvas The vector renderer to use to draw the service. Usually L.svg but setting to L.canvas contains performance benefits for large polygon layers.

... extra options

---

**imageMapLayerOptions**  
*Options for image map layer.*

**Description**

Options for image map layer.

**Usage**

imageMapLayerOptions(format = "jpg/png", f = "json", opacity = 1,  
position = "front", maxZoom = NULL, minZoom = NULL, from = NULL,  
to = NULL, bandIds = NULL, noData = NULL, noData Interpretation = NULL,  
pixelType = NULL, renderingRule = NULL, mosaicRule = NULL,  
token = NULL, proxy = NULL, useCors = TRUE, ...)

**Arguments**

**format**  
Output format of the image.

**f**  
Server response content type.

**opacity**  
Opacity of the layer. Should be a value between 0 and 1.

**position**  
Position of the layer relative to other overlays.

**maxZoom**  
Closest zoom level the layer will be displayed on the map.

**minZoom**  
Furthest zoom level the layer will be displayed on the map.

**from**  
Date When paired with to defines the time range of data to display. Requires the Image Layer to be time enabled.

**to**  
Date When paired with from defines the time range of data to display. Requires the Image Layer to be time enabled.

**bandIds**  
If there are multiple bands, you can specify which bands to export.

**noData**  
The pixel value representing no information.

**noData Interpretation**  
Interpretation of the noData setting.

**pixelType**  
Leave pixelType as unspecified, or UNKNOWN, in most exportImage use cases, unless such pixelType is desired. Possible values: C128, C64, F32, F64, S16, S32, S8, U1, U16, U2, U32, U4, U8, UNKNOWN.
renderingRule  A JSON representation of a raster function
mosaicRule    A JSON representation of a mosaic rule
token         If you pass a token in your options it will be included in all requests to the service.
proxy         URL of an ArcGIS API for JavaScript proxy or ArcGIS Resource Proxy to use for proxying requests.
useCors       If this service should use CORS when making GET requests.
...            extra options

-------------------
leaflet.esri       leaflet.esri: 'ESRI' bindings for the 'leaflet' Package.
-------------------

Description
An add-on package to the 'leaflet' package, which provides bindings for 'ESRI' services. This package allows a user to add 'ESRI' provided services such as 'MapService', 'ImageMapService', 'TiledMapService' etc. to a 'leaflet' map.

Author(s)
Bhaskar V. Karambelkar

-------------------
TiledMapLayerOptions Options for TiledMapLayer.
-------------------

Description
Options for TiledMapLayer.

Usage
tiledMapLayerOptions(correctZoomLevels = TRUE, zoomOffsetAllowance = 0.1, proxy = NULL, useCors = TRUE, token = NULL, tileOptions = NULL)

Arguments
correctZoomLevels Correct Zoom levels.
zoomOffsetAllowance If correctZoomLevels is enabled this controls the amount of tolerance for the difference at each scale level for remapping tile levels.
proxy              URL of an ArcGIS API for JavaScript proxy or ArcGIS Resource Proxy to use for proxying requests.
useCors            Dictates if the service should use CORS when making GET requests.
token              Use this token to authenticate all calls to the service.
tileOptions        Other options for tile layer. You can pass tileOptions().
Index

*Topic datasets
  esriBasemapLabels, 10
  esriBasemapLayers, 10

  addEsriBasemapLayer, 2
  addEsriClusterDependency
    (addEsriDependency), 3
  addEsriDependency, 3
  addEsriDynamicMapLayer, 3
  addEsriFeatureLayer, 4
  addEsriGeocoderDependency
    (addEsriDependency), 3
  addEsriHeatmapDependency
    (addEsriDependency), 3
  addEsriHeatmapFeatureLayer, 6
  addEsriImageMapLayer, 7
  addEsriRenderersDependency
    (addEsriDependency), 3
  addEsriTiledMapLayer, 8
  addTiles, 5, 7

dynamicMapLayerOptions, 9
  esriBasemapLabels, 10
  esriBasemapLayers, 10

  featureLayerOptions, 11
  highlightOptions, 6
  iconList, 5
  imageMapLayerOptions, 12

  labelOptions, 5
  leaflet.esri, 13
  leaflet.esri-package (leaflet.esri), 13

  makeIcon, 5
  markerClusterOptions, 5

  popupOptions, 4, 5, 7

tiledMapLayerOptions, 13
tileOptions, 2, 13

tileOptions, 2, 13