

# Package ‘geoarrowWidget’

May 8, 2026

**Title** Attach '(Geo)Arrow' and/or '(Geo)Parquet' Data to a Widget

**Version** 0.1.0

**Maintainer** Tim Appelhans <tim.appelhans@gmail.com>

**Description** Provides functions and necessary 'JavaScript' bindings to quickly transfer spatial data from R memory or remote URLs to the browser for use in interactive 'HTML' widgets created with the 'htmlwidgets' R package. Leverages 'GeoArrow' (<<https://geoarrow.org/>>) data representation for data stored in local R memory which is generally faster than traditional 'GeoJSON' by minimising the amount of copy, serialization and deserialization steps necessary for the data transfer. Furthermore, provides functionality and 'JavaScript' bindings to consume 'GeoParquet' (<<https://geoparquet.org/>>) files from remote URLs in the browser.

**License** MIT + file LICENSE

**Imports** htmltools, htmlwidgets, listviewer, nanoarrow

**Suggests** geoarrow, quarto, tinytest, wk

**VignetteBuilder** quarto

**Encoding** UTF-8

**RoxygenNote** 7.3.3

**URL** <https://github.com/r-spatial/geoarrowWidget>,  
<https://r-spatial.github.io/geoarrowWidget/>

**BugReports** <https://github.com/r-spatial/geoarrowWidget/issues>

**SystemRequirements** Quarto command line tool  
(<<https://github.com/quarto-dev/quarto-cli>>).

**NeedsCompilation** no

**Author** Tim Appelhans [cre, aut] (ORCID:  
<<https://orcid.org/0000-0002-9824-2707>>)

**Repository** CRAN

**Date/Publication** 2026-02-18 19:10:03 UTC

## Contents

attachData . . . . .	2
attachGeoarrowDependencies . . . . .	3
extJSLibs . . . . .	4
geoarrowDummyWidget . . . . .	5

<b>Index</b>	<b>7</b>
--------------	----------

---

attachData	<i>Attach data to a widget.</i>
------------	---------------------------------

---

### Description

Pipe-friendly function to attach data (any, really, not only geoarrow) to a widget created with [createWidget](#).

### Usage

```
attachData(widget, data, file, url, ...)
```

### Arguments

widget	A widget created with <a href="#">createWidget</a> .
data	a <code>nanoarrow_array_stream</code> as created with <a href="#">as_nanoarrow_array_stream</a> . If supplied, file and url are ignored.
file	A local file path to a data file to be attached to the widget. Ignored if data is supplied.
url	A URL to a file to be attached to the widget. Ignored if data or file is supplied.
...	further arguments supplied to internal methods. The most relevant argument is name which can be used to set the id of the attachment. See details and examples for further explanation.

### Details

The provided data will be attached to the page created by the widget as a `<link>`. It can then be used by some script that will fetch this data from the href. See e.g. the "Use geoarrowWidget with an existing widget" vignette of this package for an example of how to work with this data using [onRender](#) or the [source of geoarrowDummyWidget.js](#) for another, similar example.

NOTE that the `<link>` id can be controlled by supplying a name argument (via ...). This will be prepended to `<name>-geoarrowWidget-attachment`. See example below, where the name "my-data" is used to create the id "mydata-geoarrowWidget-attachment". In case no name is supplied, it defaults to the file name (without extension) that is supplied via file or url.

### Value

The widget with the data attached.

## Examples

```
library(listviewer)

wgt = jsonedit(
  list("Just some dummy text")
  , elementId = "lv-example"
)
attachData(
  wgt
  , url = "https://geoarrow-test.s3.eu-central-1.amazonaws.com/test_layer_interleaved.arrow"
  , name = "mydata"
)

## open the resulting page in the browser and inspect the page source, e.g.
## by pressing <Ctrl + u>. You should see a line like (href is shortened here):

# <link id="mydata-geoarrowWidget-attachment" rel="attachment" href="https://geoarrow-test.s3..." />
```

---

attachGeoarrowDependencies

*Attach (Geo)Arrow and/or (Geo)Parquet JavaScript dependencies to a widget.*

---

## Description

Pipe-friendly functions to attach (Geo)Arrow and/or (Geo)Parquet JavaScript dependencies to a widget created with `createWidget`.

## Usage

```
attachGeoarrowDependencies(widget)

attachGeoarrowDependency(widget)

attachArrowDependency(widget)

attachParquetWasmDependencies(widget)
```

## Arguments

widget            A widget created with `createWidget`.

## Details

Attaching the parquet-wasm JavaScript dependency differs from attaching the other dependencies. In order to enable reading `.parquet` files in the browser we declare an async function

parquet2arrow at the window level that can be used to read parquet data into an Arrow memory table in the browser. As such, attachParquetWasmDependencies() will also attach the arrow and geoarrow dependencies.

So, in the browser, we can use this as follows:

```
fetch(<(geo)parquet-url>)
  .then(pq => window.parquet2arrow(pq))
  .then(arrow_table => {

    // code to work with arrow table

  });
```

### Value

The widget including Arrow, Geoarrow and/or parquet-wasm JavaScript dependencies.

### Examples

```
library(listviewer)

wgt = jsonedit(
  list("Just some dummy text")
  , elementId = "lv-example"
)
attachGeoarrowDependencies(wgt)

## open the resulting page in the browser and inspect the page source, e.g.
## by pressing <Ctrl + u>. You should see two lines like:

# <script src="lib/apache-arrow-js-19.0.1/Arrow.es2015.min.js"></script>
# <script src="lib/geoarrow-js-0.3.2/geoarrow.umd.min.js"></script>

## !Version numbers of the JavaScript dependencies may differ!
```

---

extJSLibs

*Names and versions of external JavaScript libraries.*

---

### Description

Names and versions of the external JavaScript libraries used in geoarrowWidget.

### Usage

```
extJSLibs()
```

**Details**

See e.g. <https://cdn.jsdelivr.net/npm/apache-arrow/package.json> or <https://cdn.jsdelivr.net/npm/@geoarrow/geoarrow-js/package.json> for more details on the js dependencies.

**Value**

A named character vector with the versions of the GeoArrow and Arrow JavaScript libraries shipped with this package.

**Examples**

```
extJSLibs()
```

---

geoarrowDummyWidget    *Basic Widget to Showcase Things.*

---

**Description**

Widget used to highlight efficiency of geoarrow data transfer. Doesn't do anything useful, other than attaching data to a generated htmlwidget and printing that in the browser console.

**Usage**

```
geoarrowDummyWidget(
  message,
  file,
  url,
  dataname = "mygeoarrowdata",
  width = NULL,
  height = NULL,
  elementId = NULL
)
```

**Arguments**

message	Some text to be displayed on the generated web page.
file	A local file path to a geoarrow file to be attached to the widget.
url	A URL to a geoarrow file to be attached to the widget. Ignored if file is supplied.
dataname	The 'ATTACHINDEX' to be used for the data attachment. See <a href="#">htmlDependency</a> for details, especially the Details section on attachment.
width	Fixed width for widget (in css units). See <a href="#">createWidget</a> for details.
height	Fixed height for widget (in css units). See <a href="#">createWidget</a> for details.
elementId	Use an explicit element ID for the widget. See <a href="#">createWidget</a> for details.

**Value**

An htmlwidget with the parsed georarrow data printed to the browser console.

**Examples**

```
url = "https://georarrow-test.s3.eu-central-1.amazonaws.com/test_layer_interleaved.arrow"  
wgt = georarrowDummyWidget(url = url)  
options(viewer = NULL)  
wgt
```

# Index

`as_nanoarrow_array_stream`, [2](#)  
`attachArrowDependency`  
    (`attachGeoarrowDependencies`), [3](#)  
`attachData`, [2](#)  
`attachGeoarrowDependencies`, [3](#)  
`attachGeoarrowDependency`  
    (`attachGeoarrowDependencies`), [3](#)  
`attachParquetWasmDependencies`  
    (`attachGeoarrowDependencies`), [3](#)  
  
`createWidget`, [2](#), [3](#), [5](#)  
  
`extJSLibs`, [4](#)  
  
`geoarrowDummyWidget`, [5](#)  
  
`htmlDependency`, [5](#)  
  
`onRender`, [2](#)