

# Package ‘BrowserViz’

February 17, 2019

**Type** Package

**Title** BrowserViz: interactive R/browser graphics using websockets and JSON

**Version** 2.5.0

**Date** 2018-06-15

**Author** Paul Shannon

**Maintainer** Paul Shannon <pshannon@systemsbiology.org>

**Depends** R (>= 3.4.0), jsonlite (>= 1.5), httpuv(>= 1.4.0)

**Imports** methods, BiocGenerics

**Suggests** RUnit, BiocStyle

**Description** Interactive graphics in a web browser from R, using websockets and JSON.

**License** GPL-2

**LazyLoad** yes

**biocViews** Visualization, ThirdPartyClient

**NeedsCompilation** no

**RoxygenNote** 6.1.1

**git\_url** <https://git.bioconductor.org/packages/BrowserViz>

**git\_branch** master

**git\_last\_commit** 12a26ef

**git\_last\_commit\_date** 2019-01-17

**Date/Publication** 2019-02-17

## R topics documented:

BrowserVizClass . . . . .	2
<b>Index</b>	<b>6</b>

---

BrowserVizClass	<i>BrowserViz: a base class (for extension) and standalone example of R/Javascript interactive web browser visualization</i>
-----------------	--

---

## Description

A concrete base class for interactive R/javascript visualization tools. Derived classes obtain socket setup, status and retrieval methods for free, obscuring many complicated details.

## Usage

```
BrowserViz(portRange=10000:10100, title="BrowserViz",
           browserFile, quiet=TRUE, httpQueryProcessingFunction=NULL)
toJSON(..., auto_unbox=TRUE)
addRMessageHandler(key, functionName)
```

```
## S4 method for signature 'BrowserVizClass'
ready(obj)
## S4 method for signature 'BrowserVizClass'
send(obj, msg)
## S4 method for signature 'BrowserVizClass'
browserResponseReady(obj)
## S4 method for signature 'BrowserVizClass'
getBrowserResponse(obj)
## S4 method for signature 'BrowserVizClass'
getBrowserInfo(obj)
## S4 method for signature 'BrowserVizClass'
closeWebSocket(obj)
## S4 method for signature 'BrowserVizClass'
port(obj)
## S4 method for signature 'BrowserVizClass'
wait(obj, msecs)
## S4 method for signature 'BrowserVizClass'
getBrowserWindowTitle(obj)
## S4 method for signature 'BrowserVizClass'
setBrowserWindowTitle(obj, newTitle)
## S4 method for signature 'BrowserVizClass'
getBrowserWindowSize(obj)
## S4 method for signature 'BrowserVizClass'
roundTripTest(obj, ...)
## S4 method for signature 'BrowserVizClass'
displayHTMLInDiv(obj, htmlText, div.id)
```

## Arguments

obj	The BrowserVizClass object returned by the class constructor.
portRange	One or more consecutive integers in the range 1025-65535. A typical choice is 9000:9024. The BrowserViz class constructor will try these one at a time in

	succession until a free port is found and the connection to your web browser is established. If no open ports are found in the supplied range, an error is reported.
title	The constructor creates a new window (or a new tab, depending on how you web browser is configured). This title is displayed at the top of the window or tab.
quiet	Trace and tracking messages are written to the R console if this variable is set to FALSE.
browserFile	defaults to NA, which is interpreted as an instruction to use viz.html in the inst directory of this package. Every subclass will have its own (possibly quite complex) browserFile, containing HTML, Javascript, CSS, and calls to powerful Javascript libraries (i.e., jQuery, d3, cytoscape.js). viz.html provides Javascript endpoints of, for instance, the getBrowserWindowTitle and setBrowserWindowTitle class methods described above.
httpQueryProcessingFunction	defaults to NULL. When not NULL this function, supplied by the subclass, is called whenever an HTTP (as opposed to a websocket) request arrives. Thus the BrowserViz subclass can transfer data to the web browser using a traditional HTTP GET when that is advantageous.
msg	A name list, with four required slots: "cmd", "status", "callback", "payload". See below.
msecs	Numeric, number of milliseconds.
newTitle	A character string.
htmlText	A character string, i.e., "<h3>successful round trip of json-encoded data, length 48915.</h3>"
div.id	A character string, i.e., "bvDemoDiv"
key	A character string, the "cmd" field of the incoming four-field JSON command, used to dispatch on, so that the proper function is called.
functionName	A character string: the name of a function to which incoming web socket json commands can be dispatched.
auto_unbox	Logical, default TRUE; unboxing: do not coerce a scalar into a 1-element list, as the new (2015) jsonlite package prefers to do.
...	Arguments for our local, very slightly redefined version of toJSONN.

## Methods

In the code snippets below, obj is an instance of the BrowserVizClass.

```
BrowserViz(portRange=10000:10100, title="BrowserViz", browserFile, quiet=TRUE, httpQueryProcessingFunction)
```

Constructs a BrowserViz object. Among the several actions included are: your default web browser browses to the uri of a minimal http server embedded in BrowserViz; the browserFile is returned to the browser; the websocket connection is initialized on both ends, and the lowest numbered port in portRange.

ready(obj): returns TRUE when the R/browser websocket connection is ready for use.

port(obj): returns the actual port being used.

getBrowserInfo(obj): returns a character string describing the browser to which we are connected, using the standard W3C DOM *navigator.userAgent*.

send(obj, msg): sends a properly structured (having four fields: cmd, callback, status, payload) JSON message to the browser.

browserResponseReady(obj): returns TRUE when the asynchronous response to the last message has been received from the browser.

getBrowserResponse(obj): returns the just-received JSON-encoded, four-field response to the latest message sent to the browser.

closeWebSocket(obj): Close the websocket port now in use, making it available for reuse.

getBrowserWindowTitle(obj): Returns the title of the web page (or tab).

setBrowserWindowTitle(obj, newTitle): Sets the title of the web page or tab to which we are currently connected.

getBrowserWindowSize(object): in pixels.

roundTripTest(object, ...): Sends json-encoded data to the browser, decodes the return, which should be identical to the original.

displayHTMLInDiv(object, htmlText, div.id): intended primarily for debugging, used currently with roundTripTest to display json/character count of the data sent and returned.

... Further arguments for toJSON, typically just the variable to be encoded.

### Author(s)

Paul Shannon

### Examples

```

if(interactive()){
  library(BrowserViz)

  browserVizBrowserFile <- system.file(package="BrowserViz", "browserCode", "dist", "bvdemo.html")
  bv <- BrowserViz(browserFile=browserVizBrowserFile, quiet=TRUE)

  ## make sure everything is ready to use
  while(!ready(bv)) Sys.sleep(0.1)

  port(bv)

  ## illustrate a "low level" call. This detail is usually hidden from
  ## the user, implemented and contained (in the case of this example)
  ## in a getBrowserWindowTitle(bv) method call. This level of detail
  ## reveals what goes on behind the scenes.

  msg <- list(cmd="getWindowTitle", status="request", callback="handleResponse", payload="")
  send(bv, msg)
  while(!browserResponseReady(bv)) Sys.sleep(0.1)
  getBrowserResponse(bv)

  ## a simpler user-level approach:
  getBrowserWindowTitle(bv)

  ## set and get the windowTitle
  setBrowserWindowTitle(bv, "new title")
  getBrowserWindowTitle(bv)

  ## BrowserViz provides another information method which, like the others, will apply
  ## and maybe be of some use to derived classes
  getBrowserWindowSize(bv)

```

```
## finally, you should close BrowserViz when you are done, returning
## the port for use by other applications.
closeWebSocket(bv)
} # if interactive
```

# Index

## \*Topic **classes**

BrowserVizClass, 2

## \*Topic **methods**

BrowserVizClass, 2

addRMessageHandler (BrowserVizClass), 2

browserResponseReady (BrowserVizClass),  
2

browserResponseReady, BrowserVizClass-method  
(BrowserVizClass), 2

BrowserViz (BrowserVizClass), 2

BrowserVizClass, 2

BrowserVizClass-class  
(BrowserVizClass), 2

class:BrowserVizClass  
(BrowserVizClass), 2

closeWebSocket (BrowserVizClass), 2

closeWebSocket, BrowserVizClass-method  
(BrowserVizClass), 2

displayHTMLInDiv (BrowserVizClass), 2

displayHTMLInDiv, BrowserVizClass-method  
(BrowserVizClass), 2

fromJSON (BrowserVizClass), 2

getBrowserInfo (BrowserVizClass), 2

getBrowserInfo, BrowserVizClass-method  
(BrowserVizClass), 2

getBrowserResponse (BrowserVizClass), 2

getBrowserResponse, BrowserVizClass-method  
(BrowserVizClass), 2

getBrowserWindowSize (BrowserVizClass),  
2

getBrowserWindowSize, BrowserVizClass-method  
(BrowserVizClass), 2

getBrowserWindowTitle  
(BrowserVizClass), 2

getBrowserWindowTitle, BrowserVizClass-method  
(BrowserVizClass), 2

port (BrowserVizClass), 2

port, BrowserVizClass-method  
(BrowserVizClass), 2

ready (BrowserVizClass), 2

ready, BrowserVizClass-method  
(BrowserVizClass), 2

roundTripTest (BrowserVizClass), 2

roundTripTest, BrowserVizClass-method  
(BrowserVizClass), 2

send (BrowserVizClass), 2

send, BrowserVizClass-method  
(BrowserVizClass), 2

setBrowserWindowTitle  
(BrowserVizClass), 2

setBrowserWindowTitle, BrowserVizClass-method  
(BrowserVizClass), 2

show, BrowserVizClass-method  
(BrowserVizClass), 2

toJSON (BrowserVizClass), 2

wait (BrowserVizClass), 2

wait, BrowserVizClass-method  
(BrowserVizClass), 2