

Package ‘svKomodo’

February 20, 2015

Type Package

Version 0.9-63

Date 2015-02-05

Title SciViews GUI API - Functions to interface with Komodo Edit/IDE

Author Philippe Grosjean [aut, cre]

Maintainer Philippe Grosjean <phgrosjean@sciviews.org>

Depends R (>= 2.6.0), svMisc (>= 0.9-68)

Imports utils

Suggests svHttp, svSocket

SystemRequirements Komodo Edit (<http://www.openkomodo.com>), SciViews-K (<http://www.sciviews.org/SciViews-K>)

Description Functions to manage the GUI client, like Komodo with the SciViews-K extension

License GPL-2

URL <http://www.sciviews.org/SciViews-R>

BugReports https://r-forge.r-project.org/tracker/?group_id=194

NeedsCompilation no

Repository CRAN

Date/Publication 2015-02-05 14:32:12

R topics documented:

svKomodo-package	2
koCmd	2
koInstall	5
koRefresh	6
svTaskCallbackManager	6

Index	8
--------------	----------

svKomodo-package	<i>SciViews GUI API - Functions to manage communication with Komodo Edit/IDE</i>
------------------	--

Description

The SciViews svKomodo package provides function to communicate with SciViews Komodo, which is Komodo Edit or IDE with the SciViews-K plugin installed (see <http://www.sciviews.org/SciViews-K>).

Details

Package: svKomodo
Type: Package
Version: 0.9-63
Date: 2015-02-05
License: GPL 2 or above, at your convenience

Author(s)

Philippe Grosjean

Maintainer: Ph. Grosjean <phgrosjean@sciviews.org>

koCmd	<i>Connect to the SciViews-K (Komodo Edit/IDE) socket server and run javascript code in Komodo</i>
-------	--

Description

If Komodo Edit/IDE with the SciViews-K extension is running on your machine, you can connect to its socket server and run javascript code in it with this function.

Usage

```
koCmd(cmd, data = NULL, async = FALSE, host = getOption("ko.host"),
      port = getOption("ko.port"), kotype = getOption("ko.kotype"), timeout = 2,
      type = c("js", "rjsonp", "output"), pad = NULL, ...)
```

Arguments

cmd	the javascript you want to execute in Komodo Edit, in a character string vector.
data	if a named list, replace «<name>» in cmd for each name of the list by its component first. If a character string, replace «<data>» in cmd. If NULL (be default), do nothing to cmd before submitting it. See the last examples for using data.
async	not used yet!
host	the host where Komodo is located. Currently, only localhost is accepted. Can be changed by setting <code>options(ko.host = ...)</code> .
port	The socket port where the SciViews-K server is listening, by default, it is port 7052. Can be changed by setting <code>options(ko.port = ...)</code> .
kotype	the type of Komodo server in use. Currently (SciViews-K >= 0.9-25), it can be either "socket" or "file".
timeout	number of seconds to wait for response.
type	which type of Komodo command do we send? If <code>type = "js"</code> (by default), cmd is considered to be JavaScript code to be evaluated in Komodo. If <code>type = "rjsonp"</code> , cmd is parsed as RJson object with padding (included in a JavaScript function call) and will be evaluated as such in Komodo. if <code>type = "output"</code> , the string in cmd is considered to be some R output and will be displayed in the Komodo local R console (no evaluation).
pad	a string naming the JavaScript function to run in Komodo, with the constructed Rjson object as argument. If NULL (by default), the Rjson object is evaluated directly without padding.
...	further arguments to pass to <code>toRjson()</code> .

Details

Komodo Edit (<http://www.openkomodo.com/>) is an Open Source (MPL, GPL & LGPL) editor based on the excellent Mozilla platform and the powerful Scintilla text editor widget. It runs on many Linux distributions, on Windows and on Mac OS X. Komodo IDE is a commercial equivalent, but with many tools for developers, especially targetting languages like Perl, Tcl, Python, Ruby, etc. We are currently in the process of rewriting SciViews-R (see <http://www.sciviews.org/SciViews-R>) in a platform independent solution using Komodo with a specific extension called SciViews-K and using SciViews-R packages (`svKomodo`, `svMisc`, `svHttp`, `svSocket`, ...) for supporting interaction between Komodo and R.

`koCmd()` can only talk to Komodo if the SciViews-K socket server is installed. This server is contained in the SciViews-K extension that you can download from <http://www.sciviews.org/SciViews-K>. See Komodo documentation to know how to install this extension (drag and drop of the extension on the Komodo window works in most platforms).

We may automatize the installation from R in the future, at least on Windows and Macintosh.

Value

Returns the results of the evaluation of the javascript code in Komodo Edit/IDE if `async = FALSE`. Note that `async = TRUE` is not supported yet.

If there is an error, or cmd is an invalid javascript code, a character string containing javascript error message is returned (this is changed from version 0.9-47, previously a 'try-error' was returned).

Note

Because of serious security issues, the SciViews-K server only allows connections from local clients (running on the same computer). This limitation would be relatively easy to eliminate, but at your own risks!

Data are returned from Komodo to R by using the javascript function `sv.socket.serverWrite()`, see the examples bellow.

Author(s)

Philippe Grosjean (<phgrosjean@sciviews.org>)

See Also

[startSocketServer](#), [processSocket](#)

Examples

```
## Not run:
## Make sure you have started Komodo Edit or IDE with the SciViews-K extension
## installed on the same machine, and the socket server started and then...

## Send JavaScript commands to Komodo
## Alert box in Komodo, and then reply to R
koCmd(c('alert("Hello from R!");',
       'sv.socket.serverWrite("Hello from OpenKomodo (" + ko.interpolate.currentFilePath() + ")");'))

## Open a web page with Komodo configuration
koCmd("ko.open.URI('about:config','browser');")

## Get info from Komodo
koCmd("sv.socket.serverWrite(ko.logging.getStack());")

## Passing a large amount of data to Komodo, and then, back to R
koCmd(paste('sv.socket.serverWrite("", rep(paste(iris, collapse = "\\n"), 10), "");', sep = ""))

## It is easier to use 'data =' instead of paste() for constructing the JS command
koCmd('alert("<<<data>>>");', data = search())

## Using a named list for data to replace in the cmd
koCmd('alert("This is R version <<<major>>>.<<<minor>>>");', R.version)

## Sending incorrect javascript instruction
koCmd('nonexistingJSfunction();')
## Should return something like:
## "ReferenceError: nonexistingJSfunction is not defined"

## Sending RjsonP (Rjson with padding) instruction to Komodo
koCmd("Hello with RjsonP!", type = "rjsonp", pad = "alert")

## This is more useful to pass complex R objects to Komodo
```

```
koCmd(head(iris), type = "rjsonp", pad = "sv.socket.serverWrite")

## Send simple text (no evaluation) to the Komodo R console
koCmd("Hello again from R!", type = "output")

## End(Not run)
```

koInstall*Install and uninstall hooks for communicating with Komodo Edit/IDE*

Description

Install functions in SciViews:TempEnv and callbacks required to communicate with Komodo Edit with the SciViews-K extension (see <http://www.sciviews.org/SciViews-K>).

Usage

```
koInstall()
koUninstall()
```

Details

The minimum instruction to install the communication with Komodo/SciViews-K (so called, SciViews Komodo) is to use: `options(ko.serve = 8888); require(svKomodo)`. When the `ko.serve` option is set, `svKomodo` loads `svSocket`, starts the socket server listening to the port you have selected (8888 by default), and install the hooks and callbacks required to communicate with SciViews Komodo.

Before loading `svKomodo`, you can also set `option(ko.port = 7052)` or another port number where the Komodo SciViews-K server is listening (7052 is the default value). If the Komodo client is running on a different machine, you should also set `ko.host = "xxx.xxx.xxx.xxx"`, where `xxx.xxx.xxx.xxx` is the IP address of the Komodo client's machine, before loading `svKomodo` (note that running R and Komodo on separate machines is not supported yet, but this is a planned feature and corresponding configurations are already recognized; just, distant server is currently locked until we will build a better security mechanism in the server (SSL, TSL, ...)).

All these operations are done by Komodo if you start R from Komodo with SciViews-K extension installed.

Value

Returns nothing.

Author(s)

Philippe Grosjean (<phgrosjean@sciviews.org>)

See Also

[koCmd](#)

koRefresh	<i>Refresh Komodo interface (active object lists and R object browser)</i>
-----------	--

Description

These function manage to refresh the list of active objects in Komodo and also the data displayed in the Komodo R object browser. They should not be called directly by the end-user.

Usage

```
koRefresh(force = FALSE)
koAutoRefresh(...)
```

Arguments

force	do we force refresh, even if data have not changed?
...	any argument (ignored, but useful for addTaskCallback()).

Value

Both functions return TRUE on success.

Author(s)

Philippe Grosjean (<phgrosjean@sciviews.org>)

See Also

[koCmd](#)

svTaskCallbackManager	<i>Create task callbacks that are evaluated both from R and socket/http server</i>
-----------------------	--

Description

svTaskCallbackManager() is a copy of taskCallbackManager() in base package, as of version 2.9.2 of R. Two important differences: (1) the top task created is named SV-taskCallbackManager instead of R-taskCallbackManager, and its tasks are executed after each top-level task in R console, or after execution of non-hidden R code from the socket or http server (take care: only once per set of code, no matter the number of top-level task in the R code send by the client in the second case). All taskCallbacks defined by addTaskCallback() or taskCallbackManager\$add() from base package are not executed when code is invoked from the R socket or http server!

Usage

```
svTaskCallbackManager(handlers = list(), registered = FALSE, verbose = FALSE)
```

Arguments

handlers	this can be a list of callbacks in which each element is a list with an element named <code>f</code> which is a callback function, and an optional element named <code>data</code> which is the 5-th argument to be supplied to the callback when it is invoked. Typically this argument is not specified, and one uses <code>add</code> to register callbacks after the manager is created.
registered	a logical value indicating whether the <code>evaluate</code> function has already been registered with the internal task callback mechanism. This is usually <code>FALSE</code> and the first time a callback is added via the <code>add</code> function, the <code>evaluate</code> function is automatically registered. One can control when the function is registered by specifying <code>TRUE</code> for this argument and calling <code>addTaskCallback()</code> manually.
verbose	a logical value, which if <code>TRUE</code> , causes information to be printed to the console about certain activities this dispatch manager performs. This is useful for debugging callbacks and the handler itself.

Value

See `?taskCallbackManager` for both the returned object and how to use it.

Author(s)

Slightly modified from the original R core team's function by Ph. Grosjean <phgrosjean@sciviews.org>

See Also

[taskCallbackManager](#)

Index

*Topic **IO**

koCmd, [2](#)

svTaskCallbackManager, [6](#)

*Topic **misc**

koInstall, [5](#)

koRefresh, [6](#)

*Topic **package**

svKomodo-package, [2](#)

*Topic **utilities**

svKomodo-package, [2](#)

koAutoRefresh (koRefresh), [6](#)

koCmd, [2](#), [5](#), [6](#)

koInstall, [5](#)

koRefresh, [6](#)

koUninstall (koInstall), [5](#)

processSocket, [4](#)

startSocketServer, [4](#)

svKomodo (svKomodo-package), [2](#)

svKomodo-package, [2](#)

svTaskCallbackManager, [6](#)

taskCallbackManager, [7](#)