Package 'reactlog'

March 22, 2019

Title Reactivity Visualizer for 'shiny'

Version 1.0.0

Description Building interactive web applications with R is incredibly easy
with 'shiny'. Behind the scenes, 'shiny' builds a reactive graph that can
quickly become intertwined and difficult to debug. 'reactlog'
(Schloerke 2019) <doi:10.5281/zenodo.2591517> provides a visual insight into
that black box of 'shiny' reactivity by constructing a directed dependency
graph of the application's reactive state at any time point in a reactive
recording.

Depends R (>= 3.0.2)

Imports jsonlite (>= 0.9.16)

Suggests shiny, knitr, rmarkdown, htmltools

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

URL https://rstudio.github.io/reactlog/,
https://github.com/rstudio/reactlog,
https://community.rstudio.com/tags/reactlog

BugReports https://github.com/rstudio/reactlog/issues

VignetteBuilder knitr

Language en-US

NeedsCompilation no

Author Barret Schloerke [aut, cre] (<https://orcid.org/0000-0001-9986-114X>),
Joe Cheng [ctb],
RStudio [cph, fnd]

Maintainer Barret Schloerke <barret@rstudio.com>

Repository CRAN

Date/Publication 2019-03-22 16:40:03 UTC

R **topics documented:**

---

reactlog_render   *Reactive Log Visualizer*

Description

Provides an interactive browser-based tool for visualizing reactive dependencies and execution in
your application.

Usage

```
reactlog_render(log, session_token = NULL, time = TRUE)

reactlog_show(log, time = TRUE, ...)
```

Arguments

log	Log produced by shiny to be processed
session_token	token to be used to subset which session is displayed. Defaults to all sessions.
time	A boolean that specifies whether or not to display the time that each reactive takes to calculate a result.
...	Future parameter expansion. Currently ignored

Details

To use the reactive log visualizer, start with a fresh R session and run the command options(shiny.reactlog=TRUE);
then launch your application in the usual way (e.g. using runApp). At any time you can hit

Ctrl+F3

(or for Mac users,

Cmd+F3

) in your web browser to launch the reactive log visualization.

The reactive log visualization only includes reactive activity up until the time the report was loaded.
If you want to see more recent activity, refresh the browser.

Note that Shiny does not distinguish between reactive dependencies that "belong" to one Shiny user
session versus another, so the visualization will include all reactive activity that has taken place in
the process, not just for a particular application or session.

As an alternative to pressing

```
Ctrl/Cmd+F3
```

–for example, if you are using reactives outside of the context of a Shiny application–you can run the showReactLog function, which will generate the reactive log visualization as a static HTML file and launch it in your default browser. In this case, refreshing your browser will not load new activity into the report; you will need to call showReactLog() explicitly.

For security and performance reasons, do not enable shiny.reactlog in production environments. When the option is enabled, it's possible for any user of your app to see at least some of the source code of your reactive expressions and observers.

See Also

shiny::showReactLog

Examples

```
## Not run:
library(shiny)
library(reactlog)

# tell shiny to log reactivity
options("shiny.reactlog" = TRUE)

# run a shiny app
app <- system.file("examples/01_hello", package = "shiny")
runApp(app)

# once app has closed, display reactlog
shiny::showReactLog()

## End(Not run)
```

---

reactlog_write	*Write reactlog*

Description

Write the reactlog to a file. If a session token is provided, all reactive interactions will be subsetted to either the global session or the session provided.

Usage

```
reactlog_write(log, file = stdout(), session_token = NULL)
```

Arguments

log	produced by shiny to be written
file	location to cat the file to
session_token	Session token identifier to be used when subsetting the complete reactlog

Index