

# Package ‘shinyloadtest’

January 9, 2020

**Type** Package

**Title** Load Test Shiny Applications

**Version** 1.0.1

**Description** Assesses the number of concurrent users 'shiny' applications are capable of supporting, and for directing application changes in order to support a higher number of users. Provides facilities for recording 'shiny' application sessions, playing recorded sessions against a target server at load, and analyzing the resulting metrics.

**License** GPL-3

**Depends** R (>= 2.10)

**Encoding** UTF-8

**LazyData** true

**URL** <https://rstudio.github.io/shinyloadtest/>,  
<https://github.com/rstudio/shinyloadtest>

**BugReports** <https://github.com/rstudio/shinyloadtest/issues>

**Collate** 'enum.R' 'data.R' 'analysis.R' 'detect.R' 'auth.R'  
'make\_report.R' 'plotting.R' 'shiny-recorder.R' 'url.R'  
'util.R' 'util-pipe.R'

**Imports** R6, curl, dplyr (>= 0.8.0.1), getPass, ggplot2, glue, gtable, htmltools, httpuv (>= 1.5.2), jsonlite, lubridate, magrittr, progress, readr, rlang (>= 0.1.2), rmarkdown, scales, stringr, tibble, svglite, websocket (>= 1.0.0), xml2

**RoxygenNote** 7.0.2

**Suggests** testthat, usethis

**SystemRequirements** pandoc (>= 2.2) - <http://pandoc.org>

**NeedsCompilation** no

**Author** Alan Dipert [aut, cre],  
Barret Schloerke [aut],  
Barbara Borges [aut],  
RStudio [cph, fnd]

**Maintainer** Alan Dipert <alan@rstudio.com>

**Repository** CRAN

**Date/Publication** 2020-01-09 10:20:02 UTC

## R topics documented:

|                                |   |
|--------------------------------|---|
| load_runs . . . . .            | 2 |
| record_session . . . . .       | 3 |
| shinyloadtest_report . . . . . | 4 |
| slt_demo_data_1 . . . . .      | 5 |
| slt_demo_data_16 . . . . .     | 6 |
| slt_demo_data_4 . . . . .      | 7 |
| slt_plot . . . . .             | 8 |

|              |           |
|--------------|-----------|
| <b>Index</b> | <b>10</b> |
|--------------|-----------|

---

|           |                                      |
|-----------|--------------------------------------|
| load_runs | <i>Create Tidy Load Test Results</i> |
|-----------|--------------------------------------|

---

### Description

The shinycannon tool creates a directory of log files for each load test. This function translates one or more test result directories into a tidy data frame.

### Usage

```
load_runs(..., verbose = TRUE)
```

### Arguments

|         |  |
|---------|--|
| ...     | Key-value pairs where the key is the desired name for the test and the value is a path to the test result directory. |
| verbose | Whether or not to print progress for reading loadtest directories  |

### Value

A tidy data frame with the test result data. Each row is an event. Columns include identifiers and timing information for the event. The variables are as follows

**run** Name of the run

**session\_id** simulated session identifier, 0-based

**user\_id** simulated user identifier, 0-based

**iteration** user session identifier, 0-based

**input\_line\_number** recording line number associated with event

**event** type of the event

**start** time the event started, in seconds, relative to the time at which all simulated users were running.

**end** time the event ended, in seconds, relative to the time at which all simulated users were running

**time** event duration, in seconds

**concurrency** number of events that happened at the same time as this one

**maintenance** whether this event occurred before or after all simulated users were running

**label** event-specific text label

**json** raw message JSON and parsed JSON of the event

### Examples

```
## Not run:
load_runs(
  `1 core` = 'results/run-1/',
  `2 cores` = 'results/run-2/'
)

## End(Not run)
```

---

record\_session

*Record a Session for Load Test*

---

### Description

This function creates a **reverse proxy** at `http://host:port` (`http://127.0.0.1:8600` by default) that intercepts and records activity between your web browser and the Shiny application at `target_app_url`.

### Usage

```
record_session(
  target_app_url,
  host = "127.0.0.1",
  port = 8600,
  output_file = "recording.log",
  open_browser = TRUE
)
```

### Arguments

|                             |   |
|-----------------------------|---|
| <code>target_app_url</code> | The URL of the deployed application.  |
| <code>host</code>           | The host where the proxy will run. Usually localhost is used.   |
| <code>port</code>           | The port for the reverse proxy. Default is 8600. Change this default if port 8600 is used by another service. |
| <code>output_file</code>    | The name of the generated recording file.   |
| <code>open_browser</code>   | Whether to open a browser on the proxy (default=TRUE) or not (FALSE).   |

## Details

By default, after creating the reverse proxy, a web browser is opened automatically. As you interact with the application in the web browser, activity is written to the `output_file` (recording.log by default).

To shut down the reverse proxy and complete the recording, close the web browser tab or window.

Recordings are used as input to the `shinycannon` command-line load-generation tool which can be obtained from the [shinyloadtest documentation site](#).

## Value

Creates a recording file that can be used as input to the `shinycannon` command-line load generation tool.

## fileInput/DT/HTTP POST support

Shiny's `shiny::fileInput()` input for uploading files, the DT package, and potentially other packages make HTTP POST requests to the target application. Because POST requests can be large, they are not stored directly in the recording file. Instead, new files adjacent to the recording are created for each HTTP POST request intercepted.

The adjacent files are named after the recording with the pattern `<output_file>.post.<N>`, where `<output_file>` is the chosen recording file name and `<N>` is the number of the request.

If present, these adjacent files must be kept alongside the recording file when the recording is played back with the `shinycannon` tool.

## See Also

[shinyloadtest articles](#)

## Examples

```
## Not run:
  record_session("https://example.com/your-shiny-app/")

## End(Not run)
```

---

shinyloadtest\_report *Make shinyloadtest Report*

---

## Description

Make shinyloadtest Report

## Usage

```
shinyloadtest_report(  
  df,  
  output = "shinyloadtest_report.html",  
  duration_cutoff = c(attr(df, "recording_duration"), 60)[1],  
  http_latency_cutoff = 5,  
  max_websocket_cutoff = 20,  
  open_browser = TRUE,  
  self_contained = TRUE,  
  verbose = TRUE  
)
```

## Arguments

|                      |   |
|----------------------|---|
| df                   | data.frame returned from <a href="#">load_runs()</a>  |
| output               | File where HTML output should be saved  |
| duration_cutoff      | Cutoff value for session duration plot. Defaults to the recording duration used to simulate df or 60 seconds. |
| http_latency_cutoff  | Cutoff value for total http latency plot  |
| max_websocket_cutoff | Cutoff value for max websocket latency plot   |
| open_browser         | Whether to open the created output in the browser   |
| self_contained       | Boolean that determines if the final output should be a self contained html file                              |
| verbose              | Boolean that determines if progress output is displayed   |

## Value

The path to the report, invisibly

## Examples

```
## Not run:  
  shinyloadtest_report(slt_demo_data_1)  
  
## End(Not run)
```

---

slt\_demo\_data\_1

*Example metrics for a 1-user load test*

---

## Description

An example dataset like that returned by [load\\_runs](#), but without the json variable for portability. Contains latency data for 1323 shiny cannon events suitable for passing to [shinyloadtest\\_report](#).

**Usage**

```
slt_demo_data_1
```

**Format**

A data frame with 1323 rows and 12 variables:

**run** Name of the run

**session\_id** simulated session identifier, 0-based

**user\_id** simulated user identifier, 0-based

**iteration** user session identifier, 0-based

**input\_line\_number** recording line number associated with event

**event** type of the event

**start** time the event started, in seconds, relative to the time at which all simulated users were running.

**end** time the event ended, in seconds, relative to the time at which all simulated users were running

**time** event duration, in seconds

**concurrency** number of events that happened at the same time as this one

**maintenance** whether this event occurred before or after all simulated users were running

**label** event-specific text label

---

```
slt_demo_data_16      Example metrics for a 16-user load test
```

---

**Description**

An example dataset like that returned by [load\\_runs](#), but without the `json` variable for portability. Contains latency data for 3942 shiny cannon events suitable for passing to [shinyloadtest\\_report](#).

**Usage**

```
slt_demo_data_16
```

**Format**

A data frame with 3942 rows and 12 variables:

**run** Name of the run

**session\_id** simulated session identifier, 0-based

**user\_id** simulated user identifier, 0-based

**iteration** user session identifier, 0-based

**input\_line\_number** recording line number associated with event

**event** type of the event

**start** time the event started, in seconds, relative to the time at which all simulated users were running.

**end** time the event ended, in seconds, relative to the time at which all simulated users were running

**time** event duration, in seconds

**concurrency** number of events that happened at the same time as this one

**maintenance** whether this event occurred before or after all simulated users were running

**label** event-specific text label

---

slt\_demo\_data\_4

*Example metrics for a 4-user load test*

---

## Description

An example dataset like that returned by [load\\_runs](#), but without the `json` variable for portability. Contains latency data for 3294 shiny cannon events suitable for passing to [shinyloadtest\\_report](#).

## Usage

```
slt_demo_data_4
```

## Format

A data frame with 3294 rows and 12 variables:

**run** Name of the run

**session\_id** simulated session identifier, 0-based

**user\_id** simulated user identifier, 0-based

**iteration** user session identifier, 0-based

**input\_line\_number** recording line number associated with event

**event** type of the event

**start** time the event started, in seconds, relative to the time at which all simulated users were running.

**end** time the event ended, in seconds, relative to the time at which all simulated users were running

**time** event duration, in seconds

**concurrency** number of events that happened at the same time as this one

**maintenance** whether this event occurred before or after all simulated users were running

**label** event-specific text label

---

`slt_plot`*Plotting outputs for shinyloadtest*

---

### Description

Many different plotting routines to display different loadtest information.

### Usage

```
slt_time_boxplot(df, labels = NULL)
```

```
slt_time_concurrency(df, labels = NULL)
```

```
slt_waterfall(df, limits = c(0, max(df$concurrency, na.rm = TRUE)))
```

```
slt_hist_loadtimes(df, max_load_time = 5)
```

```
slt_user(df)
```

```
slt_session(df)
```

```
slt_session_duration(df, cutoff = c(attr(df, "recording_duration"), 60)[1])
```

```
slt_session_latency(df)
```

```
slt_http_latency(df, cutoff = 10)
```

```
slt_websocket_latency(df, cutoff = 10)
```

### Arguments

|                            |   |
|----------------------------|---|
| <code>df</code>            | data frame returned from <a href="#">load_runs</a>  |
| <code>labels</code>        | A vector of labels to include. If none are supplied, all labels will be used.                 |
| <code>limits</code>        | passed into <a href="#">scale_colour_gradientn</a>  |
| <code>max_load_time</code> | The amount of time users will wait for the page to load when first requesting the app.        |
| <code>cutoff</code>        | Where to draw a horizontal or vertical line to display a reasonable cutoff line for requests. |

### Value

A [ggplot](#) plot object



**Functions**

- `slt_time_boxplot`: Box plot of load times for each event in each run
- `slt_time_concurrency`: Time on concurrency for each event for each run
- `slt_waterfall`: Event waterfall for each session within each run
- `slt_hist_loadtimes`: Histogram of page load times
- `slt_user`: Gantt chart of event duration for each user within each run
- `slt_session`: Event gantt chart of each user session within each run
- `slt_session_duration`: Event gantt chart of fastest to slowest session times within each run
- `slt_session_latency`: Stacked bar chart of event duration for each session within each run
- `slt_http_latency`: Bar chart of total HTTP latency for each session within each run
- `slt_websocket_latency`: Bar chart of maximum calculation (websocket) latency for each session within each run

**Examples**

```
slt_session_duration(slt_demo_data_4)

slt_time_boxplot(slt_demo_data_16)
slt_time_concurrency(slt_demo_data_16)
slt_waterfall(slt_demo_data_16)
slt_hist_loadtimes(slt_demo_data_16)
slt_user(slt_demo_data_16)
slt_session(slt_demo_data_16)
slt_session_duration(slt_demo_data_16)
slt_session_latency(slt_demo_data_16)
slt_http_latency(slt_demo_data_16)
slt_websocket_latency(slt_demo_data_16)
```

# Index

## \*Topic **datasets**

- slt\_demo\_data\_1, 5
- slt\_demo\_data\_16, 6
- slt\_demo\_data\_4, 7

ggplot, 8

load\_runs, 2, 5–8

load\_runs(), 5

record\_session, 3

scale\_colour\_gradientn, 8

shinyloadtest\_report, 4, 5–7

slt\_demo\_data\_1, 5

slt\_demo\_data\_16, 6

slt\_demo\_data\_4, 7

slt\_hist\_loadtimes (slt\_plot), 8

slt\_http\_latency (slt\_plot), 8

slt\_plot, 8

slt\_session (slt\_plot), 8

slt\_session\_duration (slt\_plot), 8

slt\_session\_latency (slt\_plot), 8

slt\_time\_boxplot (slt\_plot), 8

slt\_time\_concurrency (slt\_plot), 8

slt\_user (slt\_plot), 8

slt\_waterfall (slt\_plot), 8

slt\_websocket\_latency (slt\_plot), 8