



What I've learned about running Shiny in production

Luis de Sousa <luisd@syep.co.za>

satRday Johannesburg 2019

Saturday, 6 April 2019

\$ whoami





**S
Y
E
O
P**

17



SYEOP

17



\$ agenda



Where

Learn

Why

When not to use



Where

Learn

Why

When not to use

+

Production



Define

Goals



Where

Learn

Why

When not to use

+

Production



Define

Goals

=



Challenges

Tools

Tips

Conclusion

Where to run Shiny



- R Studio Shiny Server Pro
- R Studio Connect
- ShinyApps.io
- Self Hosted

Learning about Shiny



- Documentation - <https://shiny.rstudio.com>
- Videos and written tutorials - <https://shiny.rstudio.com/tutorial>
- Blogs
 - R Studio - <https://blog.rstudio.com/categories/shiny>
 - Dean Attali - <https://deanattali.com>
- Twitter - [#rstats](https://twitter.com/rstats)

Why use Shiny



- Interactive R application
- Fast prototyping
- Build and deploy model in R
 - No code rewrite
- Open-source R package

Reasons not to use Shiny



- High throughput
 - R is an interpreted computer language
- Multi-page applications

Definition of Production



- Users relying on an environment with consequences when it's down

Goals of Production



- Uptime
- Secure
- Bug free
- Performance

Challenges to Shiny in Production



- Organisation - Proving it can be production ready
- Cultural – Team's willingness to adopt
- Technical – R developers are not traditional web or software developers

Tools: shinytest



- <https://rstudio.github.io/shinytest/>

~/R/win-library/3.5/shinytest/recorder - Shiny

Shiny Widgets Gallery

For each widget below, the Current Value(s) window displays the value that the widget provides to shinyServer. Notice that the values change as you interact with the widgets.

Action button

Action

Current Value:

```
[1] 1  
attr(,"class")  
[1] "integer"
```

See Code

Single checkbox

☐ Choice A

Current Value:

```
[1] FALSE
```

See Code

Checkbox group

☐ Choice 1
☐ Choice 2
☐ Choice 3

Current Values:

```
NULL
```

See Code

Date input

2014-01-01

Current Value:

Date range

2019-04-06 to 2019-04-06

Current Values:

File input

Browse... No file selected

Current Value:

Test event recorder

Take snapshot ?

Save script and exit test event recorder

Quit without saving

On exit, save test script as:

widgets-gallery-test


☒ Open script in editor on exit
☒ Run test script on exit


Random seed: ?


(None)

Recorded events	
Event type	Name
1 input	action
2 output-event	--
3 input	checkbox
4 output-event	--
5 input	checkboxGroup
6 output-event	--

Test event recorder

 Take snapshot ?

 Save script and exit test event recorder

 Quit without saving

On exit, save test script as:

widgets-gallery-test

☒ Open script in editor on exit

☒ Run test script on exit

Random seed: ?

(None)

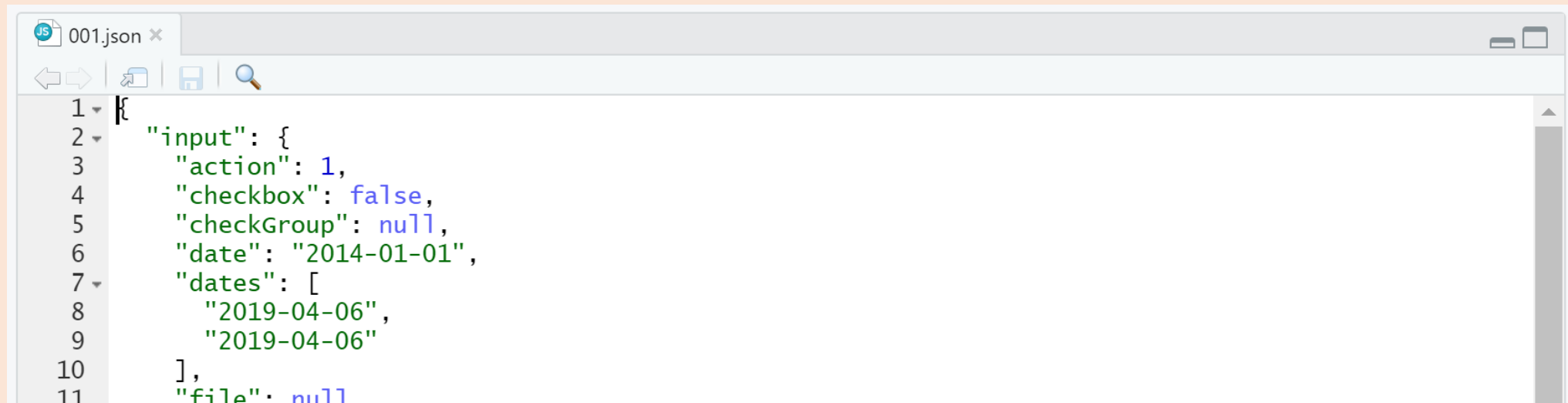
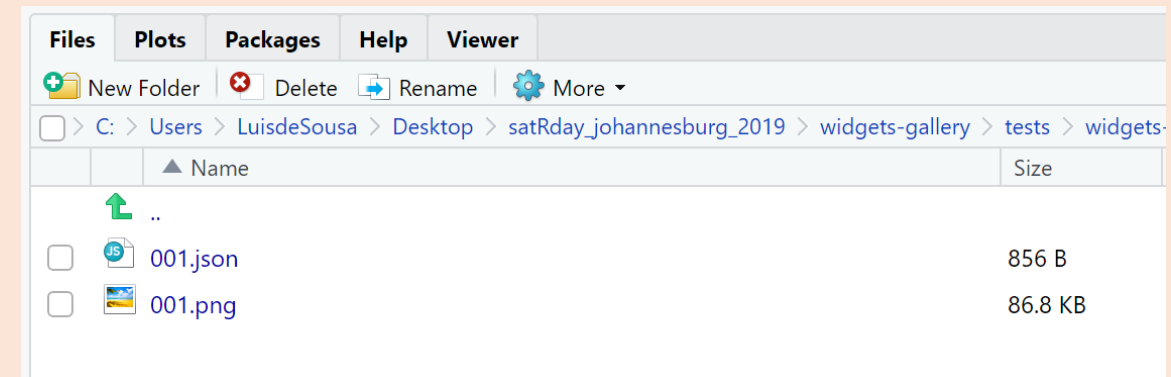
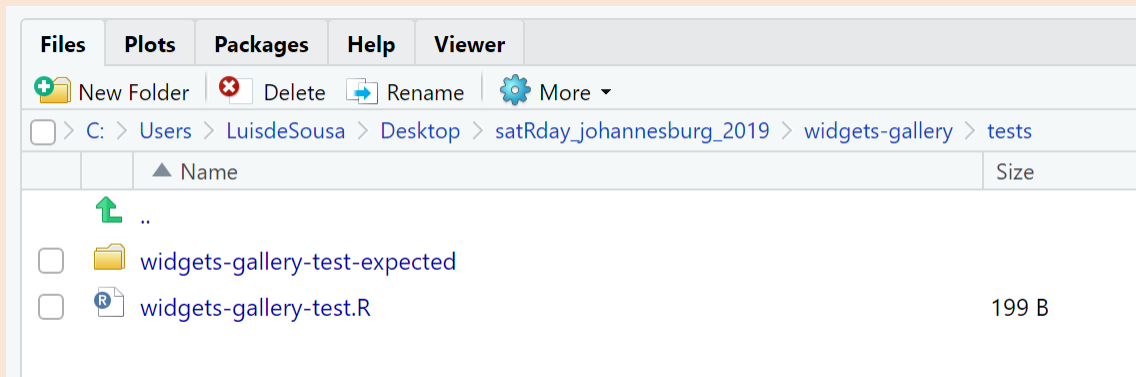
Recorded events

	Event type	Name
1	input	action
2	output-event	--
3	input	checkbox
4	output-event	--
5	input	checkboxGroup

Tools: shinytest



- <https://rstudio.github.io/shinytest/>



Tools: shinytest



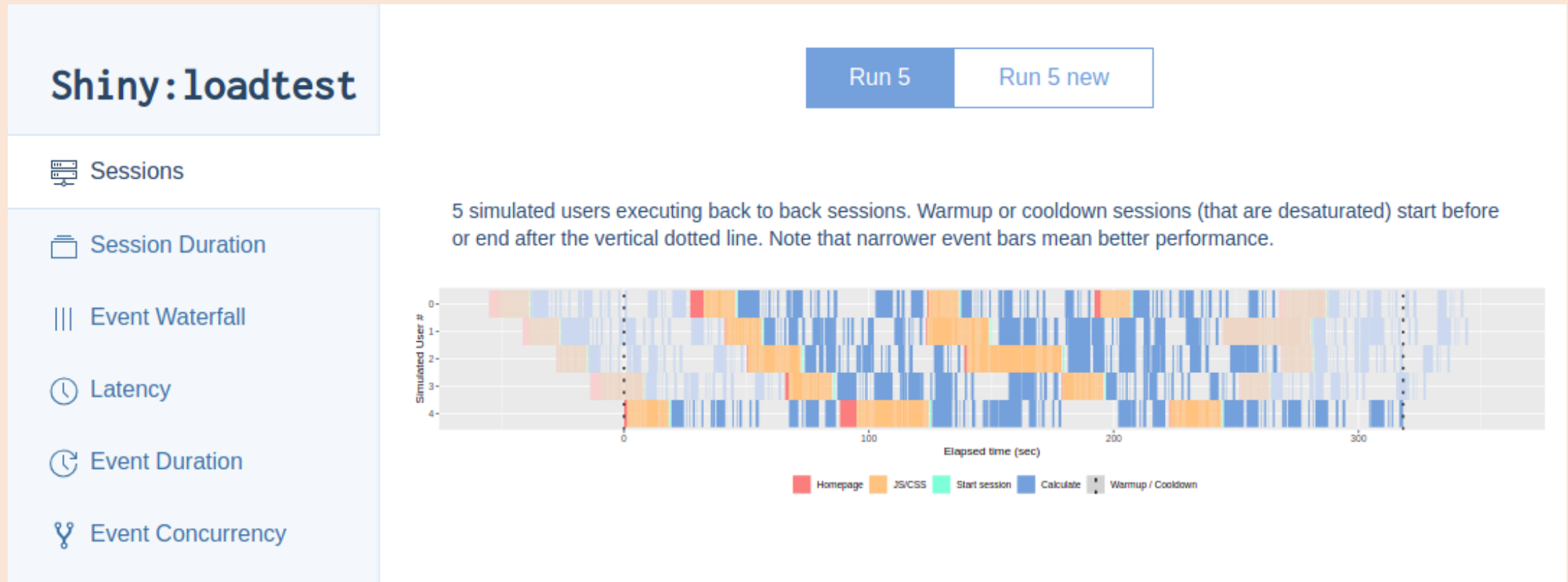
- <https://rstudio.github.io/shinytest/>

```
widgets-gallery-test.R x
1 app <- ShinyDriver$new("../")
2 app$snapshotInit("widgets-gallery-test")
3
4 app$setInputs(action = "click")
5 app$setInputs(checkbox = FALSE)
6 app$setInputs(checkGroup = character(0))
7 app$snapshot()
8
```

Tools: shinyloadtest



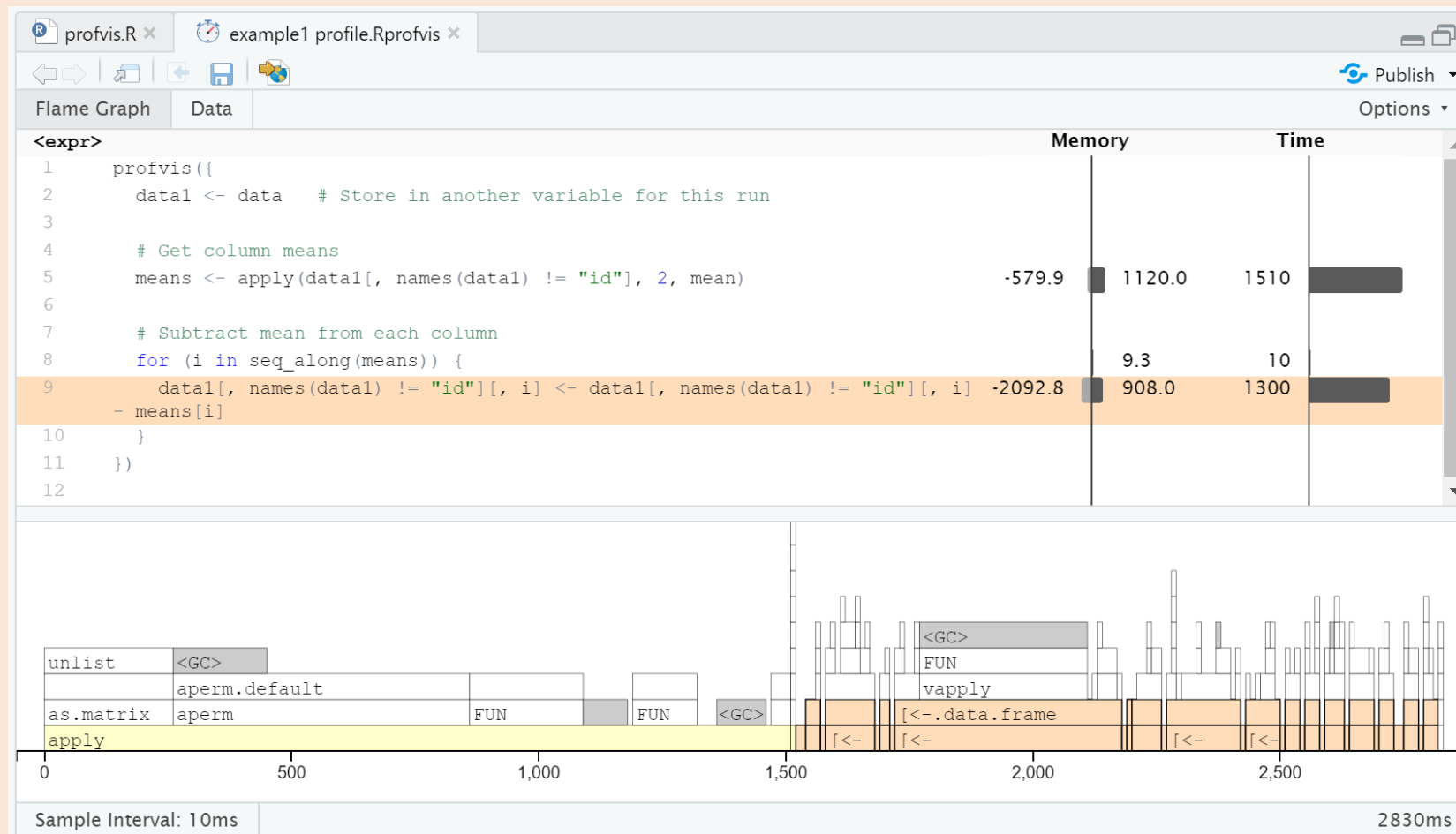
- <https://rstudio.github.io/shinyloadtest/>
- Record, Replay and Analyse



Tools: profvis



- <https://rstudio.github.io/profvis/>



Tools: Asynchronous programming



- <https://rstudio.github.io/promises/>
- “DO use promises if you have a Shiny app with long-running operations, and want to serve multiple users simultaneously.”
- “DON’T use promises to improve the performance of Shiny apps for a single user.”

Tools: Feather



- <https://github.com/wesm/feather>
- Binary columnar serialization for data frames
- Long running query that interrupts user experience

Tips: Setting up Production pipeline



- Linting
- Version control
- Code reviews and documentation
- Testing
- Deployment
 - CI / CD
 - Docker and Kubernetes
 - RStudio Connect

Tips: Filter as close to source as possible



- Move data wrangling as close to data source as possible

Tips: Plot caching



- <https://shiny.rstudio.com/articles/plot-caching.html>

```
output$plot <- renderCachedPlot(  
  {  
    # Plotting code here...  
  },  
  cacheKeyExpr = { list(input$n, dataset()) }  
)
```

Tips: Horizontal scaling



- Shiny applications can be scaled horizontally for high loads



Conclusion

- Use the tools and tips covered
- These things take time
 - Start early
 - Retrofitting is difficult
- Deploying in production is a team effort
 - Learn on the expertise of your friendly IT/Engineering resource

Resources



- https://github.com/luisdza/satRday_johannesburg_2019
- <https://resources.rstudio.com/shiny/shiny-in-production-principles-practices-and-tools-joe-cheng>
- <https://twitter.com/ColinFay/status/1085930604305235970>