

Testing Shiny applications with Shinytest

Winston Chang



RStudio::Conf 2018

2018-02-02

**Develop a
Shiny
application**

**Put it in
production**

Keep it running

Things that can change or break a Shiny application:

- Modifying your application code
- Upgrading Shiny or other packages
- Upgrading R
- System upgrades
- External data source changes or fails

Testing

Manual testing: takes a lot of time, is inconsistent, is often forgotten.

Automated testing: is really hard.

Why? Because it requires a web browser, simulating user interactions with the browser, and writing tests for graphical elements.

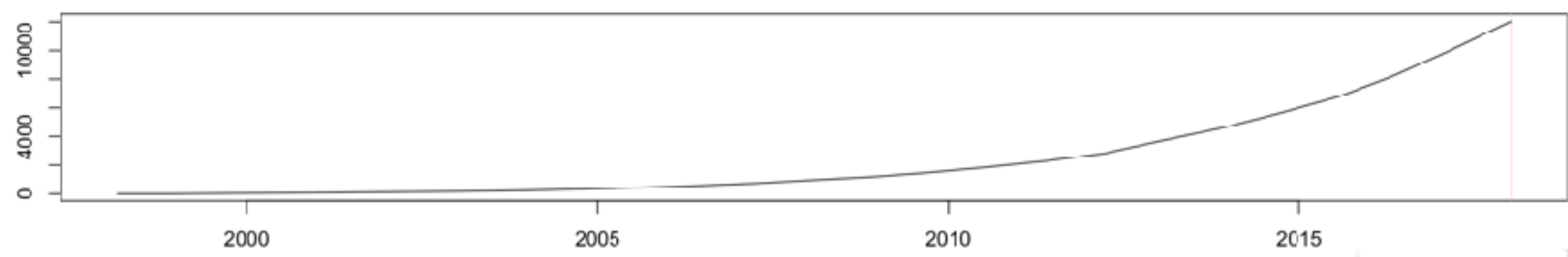
Shinytest

Snapshot-based testing for Shiny

CRAN explorer

Overview Package info

Date



☐ Log-10 scale REFRESH DATA

2018/01/10

Available packages

12056

Packages released

5

New packages

0

Packages released on this day

Shinytest procedure

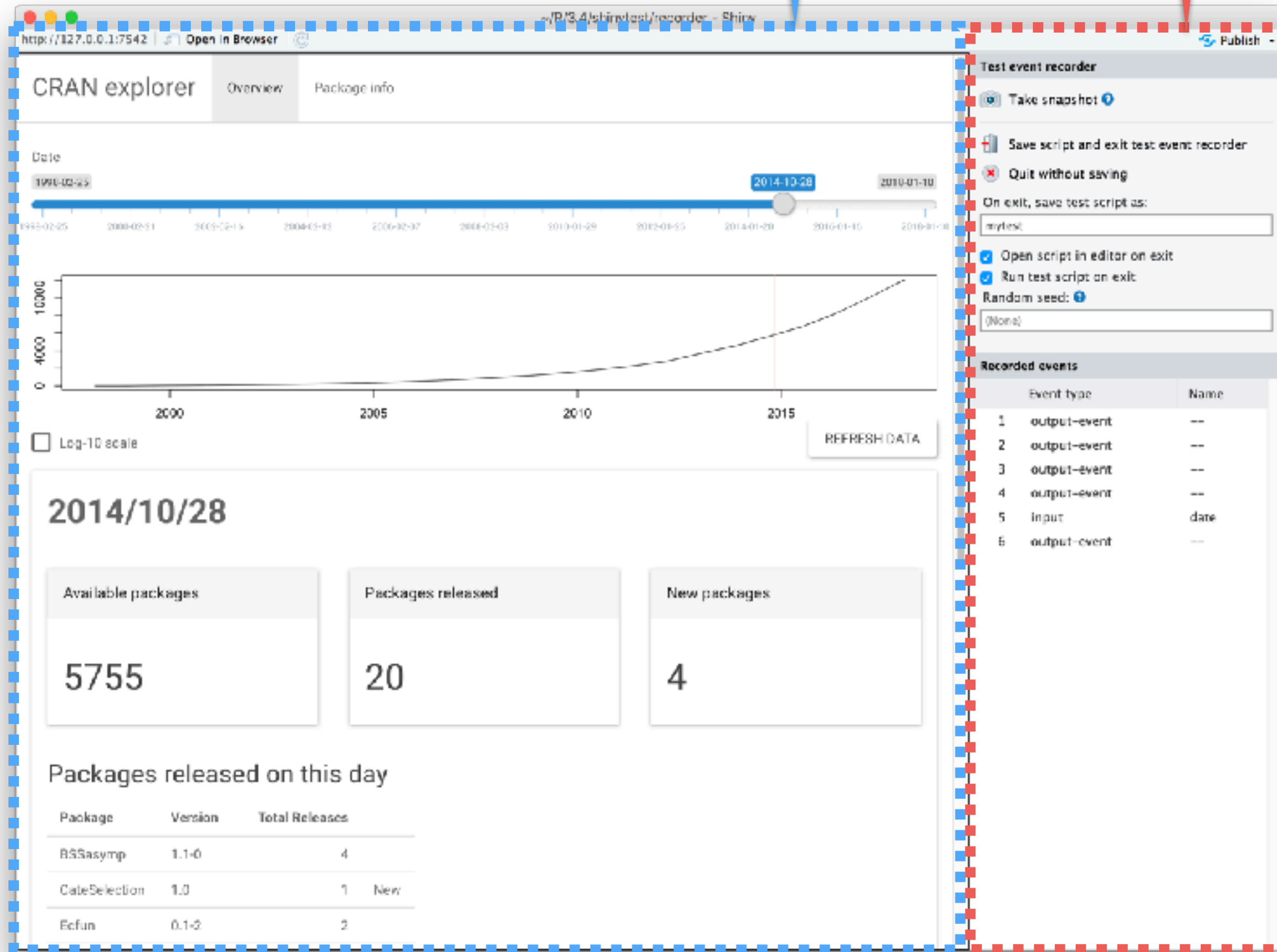
- **Create a test script**
- **Run test script to make baseline (expected) snapshots**
- **Make your changes**
- **Re-run the test script and compare snapshots**

Creating a test script

```
library(shinytest)  
recordTest("path/to/app")
```

Target app

Recorder app



Example test script

```
app <- ShinyDriver$new("../")  
app$snapshotInit("mytest")
```

```
app$snapshot()  
app$setInputs(date = "2015-11-16")  
app$setInputs(cran_timeline_log = TRUE)  
app$snapshot()
```

Snapshots

Snapshots are numbered 001, 002, etc.

Each snapshot has two files:

001.json: The state of inputs, outputs, and exported values

001.png: A screenshot of the browser

Snapshot JSON file

```
{
  "input": {
    "cran_timeline_log": true,
    "date": "2015-11-16",
    "package": "",
    "refresh": 0,
    "tabs": "Overview"
  },
  "output": {
    "cran_timeline": {
      "src": "[image data sha1: b1a8bf15fa20a7673bbbc3aa120f064]",
      "width": 962,
      "height": 160,
      ...

      "info_date": "2015/11/16",
      "info_n_new_packages_day": "4",
      "info_n_packages": "7360",
      "info_n_packages_day": "25",
      ...
    }
  }
}
```

Running tests

```
> testApp("path/to/app")
```

```
Running mytest.R
```

```
===== Comparing mytest... Passed.
```

**Success (snapshots
are as expected)**

```
> testApp("path/to/app")
```

```
Running mytest.R
```

```
===== Comparing mytest...
```

```
Differences detected between mytest-current/ and  
mytest-expected/:
```

**Maybe failure (snapshots
differ from expected)**

Name	Status
001.json	!= Files differ
001.png	!= Files differ

```
Would you like to view the differences between expected  
and current results [y/n]?
```

Difference viewer

~R/3.4/shinytest/difviewerapp - Shiny

http://127.0.0.1:7542 | Open in Browser | Publish

Differences between expected (old) and current (new) test results for cran_explorer: mytest

Update and quit | Quit

```
74 74 "info_date": "2015/11/16",
75 - "info_n_new_packages_day": "4",
75 + "info_n_new_packages_day": "5",
76 76 "info_n_packages": "7368",
77 - "info_n_packages_day": "25",
78 - "info_released_packages_table": "<table class = 'table shiny-table table- spacing-s' style = 'width:auto;'>\n<thead> <tr> <th style='text-
77 + "info_n_packages_day": "26",
78 + "info_released_packages_table": "<table class = 'table shiny-table table- spacing-s' style = 'width:auto;'>\n<thead> <tr> <th style='text-
79 79 },
80 80 "export": {
81 81
82 82 }
```

001.png | confirm

1:1 | 1:2 | 1:1 | 2:1 | Difference | Toggle | Slider

CRAN explorer

2015/11/16

Available packages: 7360

Packages released: 28

New packages: 5

Packages released on this day

Package	Version	Downloads	Downloads	Downloads
car	1.0.0	2	1	New
car	1.0.0	0	1	New
car	1.0.0	4	4	
car	1.0.0	1	1	New
car	1.0.0	7	14	
car	1.0.0	9	4	
car	1.0.0	8	1	New

What kind of changes can happen?

Benign →



Update and quit

- Minor plot rendering change
- New inputs or outputs after modifying app

Problematic →



Quit



Fix app

- App crashes or doesn't start
- Plots don't render
- Wrong output

When should I run tests?

Am I doing something that could cause the behavior of my application to change?

- Modifying your application
- Upgrading packages
- Upgrading R
- Upgrading system packages
- Changes to external data

Limitations

- Recorder is pretty good, but not perfect.
- Recorder does not capture input values from htmlwidgets (like leaflet, plotly, DT).
- Applications that use a dynamic external data source may be harder to test.

Future plans

- CRAN release very soon
- More integration with RStudio IDE
- Integration with RStudio Connect and Shiny Server Pro
 - Automatically run tests when app is deployed
 - Automatically run tests on a schedule

Thanks!

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