

Package ‘vdiff’

November 15, 2016

Title Visual Regression Testing and Graphical Diffing

Version 0.1.1

Description An extension to the 'testthat' package that makes it easy to add graphical unit tests. It provides a Shiny application to manage the test cases.

Depends R (>= 3.2.3)

Imports purrr (>= 0.2.0), R6, shiny, devtools, testthat (>= 1.0.0), svglite (>= 1.2.0), grDevices, xml2, htmlwidgets (>= 0.6), fontquiver (>= 0.2.0), gdttools (>= 0.1.2), Rcpp

Suggests ggplot2, roxygen2, rstudioapi

License GPL-3 | file LICENSE

LazyData true

RoxygenNote 5.0.1

LinkingTo Rcpp

SystemRequirements FreeType >= 2.6.0

NeedsCompilation yes

Author Lionel Henry [cre, aut],
RStudio [cph],
Carl Sutherland [aut] (jg-imagediff library),
Humble Software [cph] (jg-imagediff library),
David Hong [aut] (TwoFace library),
jQuery Foundation [cph] (jQuery library),
jQuery contributors [ctb, cph] (jQuery library; authors listed in
inst/htmlwidgets/lib/jquery-authors.txt)

Maintainer Lionel Henry <lionel@studio.com>

Repository CRAN

Date/Publication 2016-11-15 19:59:16

R topics documented:

add_dependency	2
collect_cases	2
expect_doppelganger	3
htmlwidgets	4
manage_cases	5
shinybindings	6
validate_cases	6
vdiffrAddin	7

Index 8

add_dependency	<i>Add a vdiffr dependency</i>
----------------	--------------------------------

Description

It is useful to record the version number of all the packages on which your visual test cases depend. A note containing a version number is added to the DESCRIPTION file for each dependency. Dependencies on svglite and ggplot2 are automatically added. In addition, `add_dependency()` can be called in any testthat file to manually add a dependency to a package.

Usage

```
add_dependency(deps)
```

Arguments

deps	A vector containing the names of the packages for which a dependency should be added.
------	---

collect_cases	<i>Collect cases</i>
---------------	----------------------

Description

These functions are mainly intended for internal use by `manage_cases()`. They are useful to programmatically collect cases.

Usage

```
collect_cases(package = ".", filter = NULL)
```

```
collect_new_cases(package = ".")
```

```
collect_mismatched_cases(package = ".")
```

```
collect_orphaned_cases(package = ".")
```

Arguments

package	Package description, can be path or package name. See as.package for more information.
filter	If not NULL, only tests with file names matching this regular expression will be executed. Matching will take on the file name after it has been stripped of "test-" and ".R".

Value

A cases object. `collect_new_cases()`, `collect_mismatched_cases()` and `collect_orphaned_cases()` return a filtered cases object.

See Also

[manage_cases\(\)](#)

expect_doppelganger *Does a figure look like its expected output?*

Description

If the test has never been validated yet, the test is skipped. If the test has previously been validated but `fig` does not look like its expected output, an error is issued. Use [validate_cases\(\)](#) or [manage_cases\(\)](#) to (re)validate the test.

Usage

```
expect_doppelganger(title, fig, path = NULL, ..., user_fonts = NULL)
```

Arguments

title	The figure title is used for creating the figure file names (all non-alphanumeric characters are converted to -). Also, <code>ggplot2</code> figures are appended with <code>ggtitle(title)</code> .
fig	A figure to test.
path	The path where the test case should be stored, relative to the <code>tests/figs/</code> folder. If NULL (the default), the current testthat context is used to create a sub-folder. Supply an empty string "" if you want the figures to be stored in the root folder.
...	Additional arguments passed to compare() to control specifics of comparison.
user_fonts	Passed to svglite() to make sure SVG are reproducible. Defaults to Liberation fonts for standard families and Symbola font for symbols.

Details

`fig` can be a `ggplot` object, a `recordedplot`, a function to be called, or more generally any object with a `print` method. If a `ggplot` object, a dependency for `ggplot2` is automatically added (see [add_dependency\(\)](#)).

Examples

```
ver <- gdttools::version_freetype()

if (ver >= "2.6.0") {
  disp_hist_base <- function() hist(mtcars$disp)
  expect_doppelganger("disp-histogram-base", disp_hist_base)
}

if (ver >= "2.6.0" && requireNamespace("ggplot2", quietly = TRUE)) {
  library("ggplot2")
  disp_hist_ggplot <- ggplot(mtcars, aes(dis)) + geom_histogram()
  expect_doppelganger("disp-histogram-ggplot", disp_hist_ggplot)
}
```

htmlwidgets

HTML Widgets for graphical comparison

Description

These widgets can be used at the console and embedded in a R Markdown document or Shiny application.

Usage

```
widget_toggle_(before, after, width = NULL, height = NULL)
```

```
widget_slide_(before, after, width = NULL, height = NULL)
```

```
widget_diff_(before, after, width = NULL, height = NULL)
```

```
widget_toggle(before, after, width = NULL, height = NULL)
```

```
widget_slide(before, after, width = NULL, height = NULL)
```

```
widget_diff(before, after, width = NULL, height = NULL)
```

Arguments

before	The picture that is taken as reference.
after	The picture against which the reference is compared.
width	Fixed width for widget (in css units). The default is NULL, which results in intelligent automatic sizing based on the widget's container.
height	Fixed height for widget (in css units). The default is NULL, which results in intelligent automatic sizing based on the widget's container.

Details

The regular versions take plots or functions as before and after arguments (see [expect_doppelganger\(\)](#) for details). The versions suffixed with underscores take HTML image sources. These can be paths to SVG files or inlined SVG images. Currently, `widget_diff_()` is compatible only with inlined images.

Examples

```
p1 <- function() hist(mtcars$disp)
p2 <- function() hist(mtcars$drat)

# You can also call these functions in a R Markdown document or
# in a Shiny application:
widget_toggle(p1, p2)
widget_slide(p1, p2)
widget_diff(p1, p2)
```

manage_cases

Manage visual test cases with a Shiny app

Description

Manage visual test cases with a Shiny app

Usage

```
manage_cases(package = ".", filter = NULL)
```

Arguments

package	Package description, can be path or package name. See as.package for more information.
filter	If not NULL, only tests with file names matching this regular expression will be executed. Matching will take on the file name after it has been stripped of "test-" and ".R".

See Also

[vdiffAddin\(\)](#), [collect_cases\(\)](#) and [validate_cases\(\)](#)

shinybindings *Shiny bindings for graphical comparison widgets*

Description

Output and render functions for using the Toggle, Slide and Diff widgets within Shiny applications and interactive Rmd documents. Used in `manage_cases()`.

Usage

```
toggleOutput(outputId, width = "100%", height = "400px")

renderToggle(expr, env = parent.frame(), quoted = FALSE)

slideOutput(outputId, width = "100%", height = "400px")

slideTransition(expr, env = parent.frame(), quoted = FALSE)

diffOutput(outputId, width = "100%", height = "400px")

diffTransition(expr, env = parent.frame(), quoted = FALSE)
```

Arguments

<code>outputId</code>	Output variable to read from.
<code>width, height</code>	Must be a valid CSS unit (like "100%", "400px", "auto") or a number, which will be coerced to a string and have "px" appended.
<code>expr</code>	An expression that generates a comparison widget.
<code>env</code>	The environment in which to evaluate <code>expr</code> .
<code>quoted</code>	Is <code>expr</code> a quoted expression (with <code>quote()</code>)? This is useful if you want to save an expression in a variable.

validate_cases *Cases validation*

Description

These functions are mainly intended for internal use by `manage_cases()`. They are useful to programmatically validate cases or delete orphaned cases.

Usage

```
validate_cases(cases = collect_new_cases())

delete_orphaned_cases(cases = collect_orphaned_cases())
```

Arguments

`cases` A cases object returned by one of the collecting functions such as [collect_cases\(\)](#).

See Also

[manage_cases\(\)](#)

vdiffRAddin

RStudio Addin for managing visual cases

Description

The package is detected by looking for the currently active project, then for the current folder if no project is active.

Usage

```
vdiffRAddin()
```

See Also

[manage_cases\(\)](#), [collect_cases\(\)](#) and [validate_cases\(\)](#)

Index

add_dependency, 2, 3
as.package, 3, 5

collect_cases, 2, 5, 7
collect_mismatched_cases
 (collect_cases), 2
collect_new_cases (collect_cases), 2
collect_orphaned_cases (collect_cases),
 2
compare, 3

delete_orphaned_cases (validate_cases),
 6

diffOutput (shinybindings), 6
diffTransition (shinybindings), 6

expect_doppelganger, 3, 5

htmlwidgets, 4

manage_cases, 2, 3, 5, 6, 7

renderToggle (shinybindings), 6

shinybindings, 6
slideOutput (shinybindings), 6
slideTransition (shinybindings), 6
svglite, 3

toggleOutput (shinybindings), 6

validate_cases, 3, 5, 6, 7
vdiffrAddin, 5, 7

widget_diff (htmlwidgets), 4
widget_diff_ (htmlwidgets), 4
widget_slide (htmlwidgets), 4
widget_slide_ (htmlwidgets), 4
widget_toggle (htmlwidgets), 4
widget_toggle_ (htmlwidgets), 4