

Package ‘shadowr’

April 22, 2022

Type Package

Title Selenium Plugin to Manage Multi Level Shadow Elements on Web Page

Version 0.0.1

Description Shadow Document Object Model is a web standard that offers component style and markup encapsulation.

It is a critically important piece of the Web Components story as it ensures that a component will work in any environment even if other CSS or JavaScript is at play on the page.

Custom HTML tags can't be directly identified with selenium tools, because Selenium doesn't provide any way to deal with shadow elements.

Using this plugin you can handle any custom HTML tags.

License Apache License 2.0

URL <https://github.com/ricilandolt/shadowr>

BugReports <https://github.com/ricilandolt/shadowr/issues>

Depends R (>= 3.0.0)

Imports RSelenium(>= 1.7.7), methods

Encoding UTF-8

RoxygenNote 7.1.2

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Ricardo Landolt [cre, aut],
Sushil Gupta [ctb] (shadow-automation-selenium plugin)

Maintainer Ricardo Landolt <ricardo.landolt1@gmail.com>

Repository CRAN

Date/Publication 2022-04-22 09:30:02 UTC

R topics documented:

shadow	2
shadow-class	2

Index**6**

shadow	<i>Shadow Class Constructor</i>
--------	---------------------------------

Description

This is a Class Constructor function to create a shadow class and are able to use the Selenium plugin to manage multiple levels of shadow elements on a web page.

Usage

```
shadow(driver)
```

Arguments

driver	The shadow class takes a R Selenium Remote Driver
--------	---

See Also

[shadow-class](#) or
<https://github.com/ricilandolt/shadowr> for examples

shadow-class	<i>CLASS shadow</i>
--------------	---------------------

Description

Selenium plugin to manage multiple levels of shadow elements on web page. Returning a `RSelenium::WebElement` which you can access like you used to.

Usage

```
find_element(shadowObject, css_selector, element)

## S4 method for signature 'shadow,character,missing'
find_element(shadowObject, css_selector, element)

## S4 method for signature 'shadow,character,webElement'
find_element(shadowObject, css_selector, element)

find_elements(shadowObject, css_selector, element)

## S4 method for signature 'shadow,character,missing'
find_elements(shadowObject, css_selector, element)

## S4 method for signature 'shadow,character,webElement'
```

```
find_elements(shadowObject, css_selector, element)

get_shadow_element(shadowObject, css_selector, element)

## S4 method for signature 'shadow,character,webElement'
get_shadow_element(shadowObject, css_selector, element)

get_all_shadow_element(shadowObject, css_selector, element)

## S4 method for signature 'shadow,character,webElement'
get_all_shadow_element(shadowObject, css_selector, element)

get_parent_element(shadowObject, element)

## S4 method for signature 'shadow,webElement'
get_parent_element(shadowObject, element)

get_sibling_element(shadowObject, element, css_selector)

## S4 method for signature 'shadow,webElement,character'
get_sibling_element(shadowObject, element, css_selector)

get_previous_sibling_element(shadowObject, element)

## S4 method for signature 'shadow,webElement'
get_previous_sibling_element(shadowObject, element)

get_next_sibling_element(shadowObject, element)

## S4 method for signature 'shadow,webElement'
get_next_sibling_element(shadowObject, element)

scroll_to(shadowObject, element)

## S4 method for signature 'shadow,webElement'
scroll_to(shadowObject, element)

is_checked(shadowObject, element)

## S4 method for signature 'shadow,webElement'
is_checked(shadowObject, element)

is_disabled(shadowObject, element)

## S4 method for signature 'shadow,webElement'
is_disabled(shadowObject, element)

is_visible(shadowObject, element)
```

```
## S4 method for signature 'shadow,webElement'  
is_visible(shadowObject, element)
```

Arguments

shadowObject	the shadow class
css_selector	selector string to find a web element
element	a web element

Value

Depends on the method, either a `RSelenium::WebElement` or a boolean.

Functions

- `find_element, shadow, character, missing-method`: Use this method if want single element
- `find_element, shadow, character, webElement-method`: Use this if you want to find a single elements from parent object
- `find_elements, shadow, character, missing-method`: Use this if you want to find all elements
- `find_elements, shadow, character, webElement-method`: Use this if you want to find all elements from parent object
- `get_shadow_element, shadow, character, webElement-method`: Use this if you want to find a single element from parent
- `get_all_shadow_element, shadow, character, webElement-method`: Use this if you want to find all elements from parent
- `get_parent_element, shadow, webElement-method`: Use this to get the parent element if web element
- `get_sibling_element, shadow, webElement, character-method`: Use this to get adjacent(sibling) element
- `get_previous_sibling_element, shadow, webElement-method`: Use this to get previous adjacent(sibling) element
- `get_next_sibling_element, shadow, webElement-method`: Use this to get next adjacent(sibling) element
- `scroll_to, shadow, webElement-method`: Use this to scroll to web element
- `is_checked, shadow, webElement-method`: Use this if you want to check if checkbox is selected
- `is_disabled, shadow, webElement-method`: Use this if you want to check if element is disabled
- `is_visible, shadow, webElement-method`: Use this if you want to find visibility of element

Slots

driver The shadow class takes a Remote Driver
 javascript_library the selenium plugin

See Also

<https://github.com/ricilandolt/shadowr> for examples

Examples

```
## Not run:
library(shadowr)
library(RSelenium)
remDr <- RSelenium::remoteDriver(
  remoteServerAddr = "host.docker.internal",
  port = 4445 , browser = "chrome")
remDr$open(silent = TRUE)
remDr$navigate(url)
shadow_rd <- shadow(remDr)
element <- find_element(shadow_rd, 'paper-tab[title="Settings"]')
elements <- find_elements(shadow_rd, 'paper-tab[title="Settings"]')
element$getElementText()
library(shadowr)
library(RSelenium)
remDr <- RSelenium::remoteDriver(
  remoteServerAddr = "host.docker.internal",
  port = 4445 , browser = "chrome")
remDr$open(silent = TRUE)
remDr$navigate(url)
shadow_rd <- shadow(remDr)
element <- find_element(shadow_rd, 'input[title="The name of the employee"]')
elements <- find_elements(shadow_rd, 'input[title="The name of the employee"]')
element$getElementText()
library(shadowr)
library(RSelenium)
remDr <- RSelenium::remoteDriver(
  remoteServerAddr = "host.docker.internal",
  port = 4445 , browser = "chrome")
remDr$open(silent = TRUE)
remDr$navigate(url)
shadow_rd <- shadow(remDr)
element <- find_element(shadow_rd, "properties-page#settingsPage>textarea#textarea")
element$getElementText()
## End(Not run)
```

Index

* **shadow**
 shadow, [2](#)
 shadow-class, [2](#)

find_element (shadow-class), [2](#)
find_element, shadow, character, missing-method
 (shadow-class), [2](#)
find_element, shadow, character, webElement-method
 (shadow-class), [2](#)
find_elements (shadow-class), [2](#)
find_elements, shadow, character, missing-method
 (shadow-class), [2](#)
find_elements, shadow, character, webElement-method
 (shadow-class), [2](#)

is_disabled, shadow, webElement-method
 (shadow-class), [2](#)
is_visible (shadow-class), [2](#)
is_visible, shadow, webElement-method
 (shadow-class), [2](#)

scroll_to (shadow-class), [2](#)
scroll_to, shadow, webElement-method
 (shadow-class), [2](#)
shadow, [2](#)
shadow-class, [2](#)

get_all_shadow_element (shadow-class), [2](#)
get_all_shadow_element, shadow, character, webElement-method
 (shadow-class), [2](#)
get_next_sibling_element
 (shadow-class), [2](#)
get_next_sibling_element, shadow, webElement-method
 (shadow-class), [2](#)
get_parent_element (shadow-class), [2](#)
get_parent_element, shadow, webElement-method
 (shadow-class), [2](#)
get_previous_sibling_element
 (shadow-class), [2](#)
get_previous_sibling_element, shadow, webElement-method
 (shadow-class), [2](#)
get_shadow_element (shadow-class), [2](#)
get_shadow_element, shadow, character, webElement-method
 (shadow-class), [2](#)
get_sibling_element (shadow-class), [2](#)
get_sibling_element, shadow, webElement, character-method
 (shadow-class), [2](#)

is_checked (shadow-class), [2](#)
is_checked, shadow, webElement-method
 (shadow-class), [2](#)
is_disabled (shadow-class), [2](#)