Package ‘shinySearchbar’

October 14, 2022

Title  Shiny Searchbar - An Input Widget for Highlighting Text and More

Version  1.0.0

Description  Add a searchbar widget to your 'Shiny' application. The widget quickly integrates with any existing element containing text to highlight matches. Highlighting is done with the 'JavaScript' library 'mark.js'. The widget includes buttons to cycle through multiple instances of the match and automatically scroll to the matches in an overflow element (or window). The widget also displays the total number of matches and which match is currently being cycled through. The widget is structured as a 'Bootstrap 3' input group.

URL  https://github.com/jes-n/shiny-searchbar

BugReports  https://github.com/jes-n/shiny-searchbar/issues

License  GPL-3

Encoding  UTF-8

LazyData  true

RoxygenNote  7.1.0

Collate  'demo.R' 'utils.R' 'input-searchbar.R'

Imports  shiny, jsonlite

Suggests  testthat

Depends  R (>= 2.10)

NeedsCompilation  no

Author  Jesse Norris [aut, cre],
        Julian Kühnel [ctb] (mark.js)

Maintainer  Jesse Norris <affeinated@gmail.com>

Repository  CRAN

Date/Publication  2020-06-02 09:50:02 UTC
### configurator

**Description**

All API options for mark.js v8.11.1, excluding the callback functions: each, filter, noMatch, and done.

**Usage**

configurator

**Format**

List with 15 named elements.

- **element** Element tag to wrap matches, default is "mark".
- **className** Class named appended to the match element, default is "".
- **exclude** Vector of element selectors to exclude from match, default is c().
- **accuracy** Match algorithm ("partially", "complementary", or "exactly"), default is "partially". (see mark.js API for more details).
- **synonyms** List of key-value pairs to consider equivalent, default is list().
- **ignorePunctuation** Vector of punctuation marks to ignore, default is c().
- **wildcards** Matching using wildcards such as "?" and "*" ("disabled", "enabled", or "withSpaces"), default is "disabled" (see mark.js API for more details).
- **iframes** Search within iframe elements, default is FALSE.
- **iframesTimeout** Maximum time (in ms) to wait for load before skipping an iframe element, default is 5000.
- **seperateWordSearch** Search for each space-separated word instead of the complete input, default is TRUE.
- **diacritics** Match using diacritic characters, default is TRUE.
- **acrossElements** Search for matches across elements, default is FALSE.
- **caseSensitive** Case sensitive matching, default is FALSE.
- **ignoreJoiners** Skip soft hyphen, zero width space, zero width non-joiner and zero width joiner, default is FALSE.
- **debug** Print debug information to the browser’s console, default is FALSE.

<table>
<thead>
<tr>
<th>configurator</th>
<th><strong>Configuration Options List for mark.js API</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>R topics documented:</td>
<td></td>
</tr>
<tr>
<td>configurator</td>
<td>demo</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Index</td>
<td>6</td>
</tr>
</tbody>
</table>
**demo**

**Details**

See the [mark.js API](#) for a detailed description of each option. Also see the mark.js [Configurator](#) for these options in action.

---

**demo**  

*Shiny Searchbar Demos*

---

**Description**

Run one of the shinySearchbar demos: "gallery" and "overflow".

**Usage**

```r
demo(name = "gallery", ...)
```

**Arguments**

- **name**
  Name of the demo, either "gallery" or "overflow".
- **...**
  Remaining arguments are passed to shiny::runApp, e.g. 'port', 'display.mode', etc.

**Examples**

```r
if (interactive()) {
  shinySearchbar::demo("gallery")
}
```

---

**searchbar**  

*Create the searchbar widget.*

---

**Description**

Create a textInput-like searchbar which can search through and highlight the user’s input within another element.

**Usage**

```r
searchbar(
  inputId,
  contextId,
  value = NULL,
  label = NULL,
  width = NULL,
  placeholder = NULL,
```
counter = TRUE,
cycler = TRUE,
scrollBehavior = c("smooth", "auto"),
markOpts = configurator,
quiet = FALSE
)

**Arguments**

- **inputId**: The input slot that will be used to access the value.
- **contextId**: The input slot of the element to be searched and highlighted.
- **value**: Initial value.
- **label**: Display label for the control, or NULL for no label.
- **width**: The width of the input, e.g. '400px', or '100%'; see `shiny::validateCssUnit()`.
- **placeholder**: A character string giving the user a hint as to what can be entered into the control. Internet Explorer 8 and 9 do not support this option.
- **counter**: Include a counter to display the number of matches found.
- **cycler**: Include an interface to cycle between the matches.
- **scrollBehavior**: Behavior of scrolling with cycler, either "auto" or "smooth"). The default is "smooth".
- **markOpts**: Options to be passed to mark.js; see `configurator` and the mark.js API for more details.
- **quiet**: Supress any warning related to incorrect/invalid arguments.

**Details**

The **counter** and **cycler** arguments add additional functionality to the searchbar. The **counter** shows the user how many matches were found and the **cycler** gives the user an interface to cycle through each match with automatic scrolling (using either the buttons or Enter and Shift+Enter).

The **markOpts** are a list of options to pass to the underlying mark.js API, which handles finding and highlighting the matches in element identified by **contextId**.

**Value**

A searchbar input control that can be added to a UI definition.

**See Also**

- `updateMarkOptions`, `configurator`

**Examples**

```r
if (interactive()) {
  ui <- fluidPage(
    searchbar("sb", "text"),
    textOutput("text")
  )
}
updateMarkOptions

server <- function(input, output) {
  output$text <- renderText("Hello world!"")
}
s SHINYApp(ui, server)

updateMarkOptions Update mark.js Options for an Initialized Searchbar Widget

Description
Only the options explicitly passed with this function are updated, the previous options are left unchanged.

Usage
updateMarkOptions(
  inputId, 
  markOpts, 
  session = shiny::getDefaultReactiveDomain(), 
  quiet = FALSE 
)

Arguments
inputId The input slot of the initialized searchbar widget.
markOpts Options to update for mark.js API.
session The session object passed to function given to shinyServer.
quiet Supress any warning related to incorrect/invalid arguments.

Value
No return value, called to update markOpts of the input control.

See Also
configurator
Index

* datasets
  configurator, 2

configurator, 2, 4, 5

demo, 3

searchbar, 3

updateMarkOptions, 4, 5