

Package ‘reactR’

February 22, 2021

Type Package

Title React Helpers

Version 0.4.4

Date 2021-02-20

Maintainer Kent Russell <kent.russell@timelyportfolio.com>

Description Make it easy to use 'React' in R with 'htmlwidget' scaffolds, helper dependency functions, an embedded 'Babel' 'transpiler', and examples.

URL <https://github.com/react-R/reactR>

BugReports <https://github.com/react-R/reactR/issues>

License MIT + file LICENSE

LazyData TRUE

Encoding UTF-8

Imports htmltools

Suggests htmlwidgets (>= 1.5.3), rmarkdown, shiny, V8, knitr, usethis, jsonlite

RoxygenNote 7.1.1

VignetteBuilder knitr

NeedsCompilation no

Author Facebook Inc [aut, cph] (React library in lib, <https://reactjs.org/>; see AUTHORS for full list of contributors), Michel Weststrate [aut, cph] (mobx library in lib, <https://github.com/mobxjs>), Kent Russell [aut, cre] (R interface), Alan Dipert [aut] (R interface)

Repository CRAN

Date/Publication 2021-02-22 14:50:06 UTC

R topics documented:

| | |
|--------------------------------------|---|
| babel_transform | 2 |
| component | 3 |
| createReactShinyInput | 3 |
| html_dependency_corejs | 4 |
| html_dependency_mobx | 5 |
| html_dependency_react | 6 |
| html_dependency_reacttools | 7 |
| React | 7 |
| reactMarkup | 8 |
| scaffoldReactShinyInput | 9 |
| scaffoldReactWidget | 9 |

| | |
|--------------|-----------|
| Index | 10 |
|--------------|-----------|

| | |
|-----------------|----------------------------------|
| babel_transform | <i>Transform Code with Babel</i> |
|-----------------|----------------------------------|

Description

Helper function to use V8 with Babel so we can avoid a JSX transformer with using reactR.

Usage

```
babel_transform(code = "")
```

Arguments

| | |
|------|-----------|
| code | character |
|------|-----------|

Value

transformed character

Examples

```
## Not run:
library(reactR)
babel_transform('<div>react div</div>')

## End(Not run)
```

| | |
|-----------|---------------------------------|
| component | <i>Create a React component</i> |
|-----------|---------------------------------|

Description

Create a React component

Usage

```
component(name, varArgs = list())
```

Arguments

| | |
|---------|---|
| name | Name of the React component, which must start with an upper-case character. |
| varArgs | Attributes and children of the element to pass along to tag as varArgs. |

Value

An [htmltools tag](#) object

Examples

```
component("ParentComponent",  
  list(  
    x = 1,  
    y = 2,  
    component("ChildComponent"),  
    component("OtherChildComponent")  
  )  
)
```

| | |
|-----------------------|-----------------------------------|
| createReactShinyInput | <i>Create a React-based input</i> |
|-----------------------|-----------------------------------|

Description

Create a React-based input

Usage

```
createReactShinyInput(  
  inputId,  
  class,  
  dependencies,  
  default = NULL,  
  configuration = list(),  
  container = htmltools::tags$div  
)
```

Arguments

| | |
|---------------|---|
| inputId | The input slot that will be used to access the value. |
| class | Space-delimited list of CSS class names that should identify this input type in the browser. |
| dependencies | HTML dependencies to include in addition to those supporting React. Must contain at least one dependency, that of the input's implementation. |
| default | Initial value. |
| configuration | Static configuration data. |
| container | Function to generate an HTML element to contain the input. |

Value

Shiny input suitable for inclusion in a UI.

Examples

```
myInput <- function(inputId, default = "") {  
  # The value of createReactShinyInput should be returned from input constructor functions.  
  createReactShinyInput(  
    inputId,  
    "myinput",  
    # At least one htmlDependency must be provided -- the JavaScript implementation of the input.  
    htmlDependency(  
      name = "my-input",  
      version = "1.0.0",  
      src = "www/mypackage/myinput",  
      package = "mypackage",  
      script = "myinput.js"  
    ),  
    default  
  )  
}
```

html_dependency_corejs

Shim Dependency for React in RStudio Viewer

Description

Add this first for 'React' to work in RStudio Viewer.

Usage

```
html_dependency_corejs()
```

Value

[htmlDependency](#)

html_dependency_mobx *Dependencies for 'mobx'*

Description

Add JavaScript 'mobx' and 'mobx-react' dependency. When using with 'react', the order of the dependencies is important, so please add `html_dependency_react()` before `html_dependency_mobx()`.

Usage

```
html_dependency_mobx(react = TRUE)
```

Arguments

`react` logical to add react 'mobx' dependencies.

Value

[htmlDependency](#)

Examples

```
if(interactive()) {  
  
  library(htmltools)  
  library(reactR)  
  
  browsable(  
    tagList(  
      html_dependency_mobx(react = FALSE),  
      div(id="test"),  
      tags$script(HTML(  
        "  
        var obs = mobx.observable({val: null})  
        mobx.autorun(function() {  
          document.querySelector('#test').innerText = obs.val  
        })  
        setInterval(  
          function() {obs.val++},  
          1000  
        )  
        "  
      ))  
    )  
  )  
}
```

Not run:
use with react
library(htmltools)

```
library(reactR)

browsable(
  tagList(
    html_dependency_react(),
    html_dependency_mobx(),
    div(id="test"),
    tags$script(HTML(babel_transform(
      "
      var obs = mobx.observable({val: null})
      var App = mobxReact.observer((props) => <div>{props.obs.val}</div>)

      ReactDOM.render(<App obs = {obs}/>, document.querySelector('#test'))

      setInterval(
        function() {obs.val++},
        1000
      )
      "
    )))
)

## End(Not run)
```

html_dependency_react *Dependencies for React*

Description

Add JavaScript 'React' dependency. For this to work in RStudio Viewer, also include [html_dependency_corejs](#).

Usage

```
html_dependency_react(offline = TRUE)
```

Arguments

offline logical to use local file dependencies. If FALSE, then the dependencies use the Facebook cdn as its src. To use with JSX see [babel_transform](#).

Value

[htmlDependency](#)

Examples

```

library(reactR)
library(htmltools)

tagList(
  tags$script(
    "
      ReactDOM.render(
        React.createElement(
          'h1',
          null,
          'Powered by React'
        ),
        document.body
      )
    "
  ),
  #add core-js first to work in RStudio Viewer
  html_dependency_corejs(),
  html_dependency_react() #offline=FALSE for CDN
)

```

html_dependency_reacttools

Adds window.reactR.exposeComponents and window.reactR.hydrate

Description

Adds window.reactR.exposeComponents and window.reactR.hydrate

Usage

```
html_dependency_reacttools()
```

Value

[htmlDependency](#)

React

React component builder.

Description

React is a syntactically-convenient way to create instances of React components that can be sent to the browser for display. It is a list for which [extract methods](#) are defined, allowing object creation syntax like `React$MyComponent(x = 1)` where `MyComponent` is a React component you have exposed to Shiny in JavaScript.

Usage

React

Format

An object of class `react_component_builder` of length 0.

Details

Internally, the [component](#) function is used to create the component instance.

Examples

```
# Create an instance of ParentComponent with two children,  
# ChildComponent and OtherChildComponent.  
React$ParentComponent(  
  x = 1,  
  y = 2,  
  React$ChildComponent(),  
  React$OtherChildComponent()  
)
```

reactMarkup

Prepare data that represents a single-element character vector, a React component, or an `htmltools` tag for sending to the client.

Description

Tag lists as returned by `htmltools::tagList` are not currently supported.

Usage

```
reactMarkup(tag)
```

Arguments

tag character vector or React component or [tag](#)

Value

A reactR markup object suitable for being passed to [createWidget](#) as widget instance data.

`scaffoldReactShinyInput`*Create implementation scaffolding for a React.js-based Shiny input.*

Description

Add the minimal code required to implement a React.js-based Shiny input to an R package.

Usage

```
scaffoldReactShinyInput(name, npmPkgs = NULL, edit = interactive())
```

Arguments

| | |
|----------------------|---|
| <code>name</code> | Name of input |
| <code>npmPkgs</code> | Optional NPM packages upon which this input is based which will be used to populate <code>package.json</code> . Should be a named list of names to versions . |
| <code>edit</code> | Automatically open the input's source files after creating the scaffolding. |

Note

This function must be executed from the root directory of the package you wish to add the input to.

`scaffoldReactWidget`*Create implementation scaffolding for a React.js-based HTML widget*

Description

Add the minimal code required to implement a React.js-based HTML widget to an R package.

Usage

```
scaffoldReactWidget(name, npmPkgs = NULL, edit = interactive())
```

Arguments

| | |
|----------------------|--|
| <code>name</code> | Name of widget |
| <code>npmPkgs</code> | Optional NPM packages upon which this widget is based which will be used to populate <code>package.json</code> . Should be a named list of names to versions . |
| <code>edit</code> | Automatically open the widget's JavaScript source file after creating the scaffolding. |

Note

This function must be executed from the root directory of the package you wish to add the widget to.

Index

* **datasets**

React, [7](#)

babel_transform, [2](#), [6](#)

component, [3](#), [8](#)

createReactShinyInput, [3](#)

createWidget, [8](#)

extract methods, [7](#)

html_dependency_corejs, [4](#), [6](#)

html_dependency_mobx, [5](#)

html_dependency_react, [6](#)

html_dependency_reacttools, [7](#)

htmlDependency, [4–7](#)

React, [7](#)

reactMarkup, [8](#)

scaffoldReactShinyInput, [9](#)

scaffoldReactWidget, [9](#)

tag, [3](#), [8](#)