

# Package: dragulaR (via r-universe)

October 11, 2026

**Type** Package

**Title** Drag and Drop Elements in 'Shiny' using 'Dragula Javascript Library'

**Version** 0.3.3

**Maintainer** Zygmunt Zawadzki <zygmunt@zstat.pl>

**Description** Move elements between containers in 'Shiny' without explicitly using 'JavaScript'. It can be used to build custom inputs or to change the positions of user interface elements like plots or tables.

**License** GPL-2

**Encoding** UTF-8

**LazyData** TRUE

**Depends** htmlwidgets, shiny

**Imports** shinyjs

**Suggests** covr, testthat

**BugReports** <https://github.com/zzawadz/dragulaR/issues>

**URL** <https://dragular.zstat.pl/>

**Config/roxygen2/version** 8.1.0

**Config/pak/sysreqs** cmake make libuv1-dev zlib1g-dev

**Repository** <https://zzawadz.r-universe.dev>

**Date/Publication** 2026-09-11 10:01:37 UTC

**RemoteUrl** <https://github.com/zzawadz/dragular>

**RemoteRef** HEAD

**RemoteSha** 12ecae7d1158d271be713134fd3138dc319f0da8

## Contents

|               |   |
|---------------|---|
| dragula       | 2 |
| dragulaOutput | 3 |
| dragulaValue  | 3 |
| useDragulajs  | 4 |

**Index****5**


---

|         |                                        |
|---------|----------------------------------------|
| dragula | <i>Register containers to dragula.</i> |
|---------|----------------------------------------|

---

**Description**

Create dragula instace to allow moving around elements of the registered containers.

**Usage**

```
dragula(x, ...)
```

**Arguments**

|                  |                                                                                                                                                                                                                                                                                                                                                                                                                          |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>x</code>   | vector of containers ids. Their's elements will become draggable.                                                                                                                                                                                                                                                                                                                                                        |
| <code>...</code> | additonal arguments passed to dragula JS as options. E.g. <code>id</code> will be an id to read from in shiny. Additional shortcut options: <ul style="list-style-type: none"> <li>• <code>copyOnly</code>: container id from which elements can only be copied (not moved)</li> <li>• <code>maxItems</code>: named list specifying maximum items per container, e.g. <code>maxItems = list(Model = 3)</code></li> </ul> |

**Value**

Return `htmlWidget`. Should be used only inside shiny ui.

**Examples**

```
if(interactive()) {
  # Basic example
  path <- system.file("apps/example01-dragula", package = "dragulaR")
  runApp(path, display.mode = "showcase")

  # Example with maxItems (limits Model container to 3 items)
  path <- system.file("apps/example08-max-items", package = "dragulaR")
  runApp(path, display.mode = "showcase")
}

# Create dragula with maxItems limit
## Not run:
dragula(c("Available", "Model"), maxItems = list(Model = 3))

## End(Not run)
```

---

|               |                        |
|---------------|------------------------|
| dragulaOutput | <i>Dragula widget.</i> |
|---------------|------------------------|

---

**Description**

Create dragula widget.

**Usage**

```
dragulaOutput(outputId)

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

**Arguments**

|          |                                                                                                             |
|----------|-------------------------------------------------------------------------------------------------------------|
| outputId | output variable to read from.                                                                               |
| expr     | An expression that generates a dragula object.                                                              |
| env      | The environment in which to evaluate expr.                                                                  |
| quoted   | Is expr a quoted expression (with quote())? This is useful if you want to save an expression in a variable. |

**Value**

dragulaOutput returns a Shiny tag list for the UI. renderDragula returns a Shiny render function for the server.

**Examples**

```
if(interactive()) {
  path <- system.file("apps/example02-input", package = "dragulaR")
  runApp(path, display.mode = "showcase")
}
```

---

|              |                                     |
|--------------|-------------------------------------|
| dragulaValue | <i>Format dragula input object.</i> |
|--------------|-------------------------------------|

---

**Description**

This is a utility function for better formatting dragula's input. It extracts the container names and their element order from the raw input object.

**Usage**

```
dragulaValue(x)
```

**Arguments**

x                      dragula input.

**Value**

A named list where names are container IDs and values are character vectors of element identifiers (from the drag attribute).

**Examples**

```
## Not run:
# Example call:
dragulaValue(input$dragula)

## End(Not run)
```

---

useDragulajs

---

*Register dragulaR's js functions for refreshing dragula object.*


---

**Description**

This function enables the js\$refreshDragulaR() JavaScript function in Shiny, which should be called after dynamically adding elements to a dragula container.

**Usage**

```
useDragulajs()
```

**Value**

A Shiny tag list that registers the JavaScript extension.

**Examples**

```
## Not run:
# See example for more details
library(dragulaR)
runApp(
  system.file("apps/example05-dragula-dynamic-elements", package = "dragulaR"),
  display.mode = "showcase")

## End(Not run)
```

# Index

`dragula`, [2](#)

`dragulaOutput`, [3](#)

`dragulaValue`, [3](#)

`renderDragula (dragulaOutput)`, [3](#)

`useDragulajs`, [4](#)