

# Package: rMosaic (via r-universe)

June 25, 2026

**Type** Package

**Title** R Bindings to the 'Mosaic' Visualization Framework

**Version** 0.1.3

**Description** Provides R bindings for 'Mosaic', a declarative grammar for linked, data-driven visualizations backed by 'DuckDB'. The package supplies 'htmlwidgets' and 'shiny' helpers for rendering 'Mosaic' specifications from R, including optional browser-side 'DuckDB-WASM' execution and selection export helpers for exploratory workflows.

**License** MIT + file LICENSE

**URL** <https://github.com/TiRizvanov/rMosaic>

**BugReports** <https://github.com/TiRizvanov/rMosaic/issues>

**Encoding** UTF-8

**Language** en-US

**RoxygenNote** 7.3.3

**Depends** R (>= 4.1.0)

**Imports** htmlwidgets (>= 1.5.4), shiny (>= 1.7.0), jsonlite (>= 1.8.0), yaml (>= 2.3.0), DBI (>= 1.1.0), duckdb (>= 1.4.0), stats

**Suggests** arrow (>= 12.0.0), knitr, rmarkdown, rstudioapi

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Timur Rizvanov [aut, cre], Edward C. Ruiz [aut], Ruben Dries [aut, rev], Dries Lab [fnd] (Host laboratory and funding source), Boston University [fnd] (Undergraduate Research Opportunities Program (UROP))

**Maintainer** Timur Rizvanov <timurr@bu.edu>

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

**Date/Publication** 2026-06-25 15:40:02 UTC

**RemoteUrl** <https://github.com/cran/rMosaic>

**RemoteRef** HEAD

**RemoteSha** d56838ab9639332a7d26994527e881b994af1574

## Contents

get_mosaic_selection . . . . .	2
list_mosaic_selections . . . . .	3
mosaic . . . . .	3
mosaicOutput . . . . .	4
renderMosaic . . . . .	5
runMosaicApp . . . . .	5
runMosaicExport . . . . .	6
runMosaicWithExport . . . . .	7
<b>Index</b>	<b>8</b>

---

get\_mosaic\_selection    *Retrieve a stored Mosaic selection*

---

### Description

After pressing "Import Selection" in a `runMosaicWithExport` app, the selected data are stored inside the package under the name printed in the status bar (e.g. "mosaic\_sel\_1"). Use this function to retrieve a selection by that name.

### Usage

```
get_mosaic_selection(name)
```

### Arguments

name                    Character scalar: the variable name returned by the import button (e.g. "mosaic\_sel\_1").

### Value

The stored data. frame, or NULL with a warning if the name is not found.

### See Also

[list\\_mosaic\\_selections](#)

---

`list_mosaic_selections`*List all stored Mosaic selections*

---

**Description**

Returns the names of all selections currently held in the package store.

**Usage**

```
list_mosaic_selections()
```

**Value**

Character vector of selection names.

**See Also**

[get\\_mosaic\\_selection](#)

---

`mosaic`*Render a Mosaic visualization in Shiny*

---

**Description**

Renders a Mosaic visualization using the provided specification and data.

**Usage**

```
mosaic(  
  spec,  
  specType = c("auto", "json", "yaml", "esm"),  
  data = NULL,  
  backend = c("r", "wasm"),  
  data_transport = c("auto", "file", "inline"),  
  data_dir = NULL,  
  width = NULL,  
  height = NULL  
)
```

**Arguments**

spec	JSON/YAML (as R list, text, or file) or ESM JS code (text or file).
specType	One of "auto" (default), "json", "yaml", or "esm".
data	Named list of data.frames to register in DuckDB.
backend	Database backend: "r" (default) for R DuckDB or "wasm" for browser WASM DuckDB.
data_transport	How 'backend = "wasm"' input tables are delivered to the browser. "auto" uses "file" when 'data_dir' is supplied and otherwise falls back to "inline" for portable widgets; "inline" keeps the row-JSON path; "file" writes Arrow IPC files to 'data_dir' and registers them in DuckDB-WASM by URL.
data_dir	Directory for "file" transport. Serve or save the widget from the same directory so relative URLs resolve.
width	CSS or pixel width (e.g. "100%", "600px", or numeric).
height	CSS or pixel height.

**Value**

An htmlwidget that renders the Mosaic visualization.

---

mosaicOutput	<i>Shiny output for Mosaic widget</i>
--------------	---------------------------------------

---

**Description**

Use this in the UI to create a placeholder for the Mosaic visualization.

**Usage**

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

**Arguments**

outputId	output variable name
width, height	CSS dimensions (e.g. '100%', '400px') for the container.

**Value**

A Shiny UI output element for a Mosaic htmlwidget.

---

renderMosaic	<i>Shiny render function for Mosaic</i>
--------------	---

---

**Description**

Use this in the server to render the Mosaic visualization to the output.

**Usage**

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

**Arguments**

expr	An expression that generates a call to <code>mosaic()</code> .
env	The environment in which to evaluate expr.
quoted	Is expr a quoted expression (with <code>quote()</code> )? This is useful if you want to save an expression in a variable.

**Value**

A Shiny render function for use in a server output assignment.

---

runMosaicApp	<i>Run a Mosaic Shiny App in the RStudio Viewer</i>
--------------	---

---

**Description**

Launches a Shiny application displaying the Mosaic visualization.

**Usage**

```
runMosaicApp(  
  spec,  
  specType = c("auto", "json", "yaml", "esm"),  
  data,  
  backend = c("r", "wasm"),  
  title = NULL,  
  width = "100%",  
  height = "600px"  
)
```

**Arguments**

spec	JSON/YAML (as R list, text, or file) or ESM JS code (text or file).
specType	One of "auto" (default), "json", "yaml", or "esm".
data	Named list of data.frames to register in DuckDB.
backend	Database backend: "r" (default) for R DuckDB or "wasm" for browser WASM DuckDB.
title	Optional page title
width	CSS or pixel width (e.g. "100%", "600px", or numeric).
height	CSS or pixel height.

**Value**

A Shiny application object.

---

runMosaicExport	<i>Run a Mosaic app and export brush/query selections back to R</i>
-----------------	---

---

**Description**

Runs a Mosaic app and allows exporting selections to R.

**Usage**

```
runMosaicExport(
  spec,
  specType = "auto",
  data,
  title = NULL,
  width = "100%",
  height = "600px",
  selection_env = NULL
)
```

**Arguments**

spec	JSON/YAML (as R list, text, or file) or ESM JS code (text or file).
specType	One of "auto" (default), "json", "yaml", or "esm".
data	Named list of data.frames to register in DuckDB.
title	Optional page title
width	CSS or pixel width (e.g. "100%", "600px", or numeric).
height	CSS or pixel height.
selection_env	where to store extracted selections

**Value**

A Shiny application object. When the user imports a selection, the selected rows are assigned into `selection_env` as `mosaic_sel_<n>`.

---

`runMosaicWithExport` *Run a Mosaic Shiny App with selection export*

---

**Description**

Launches a Shiny application that allows selecting data points and exporting them to an R environment.

**Usage**

```
runMosaicWithExport(  
  spec,  
  specType = c("auto", "json", "yaml", "esm"),  
  data,  
  title = NULL,  
  width = "100%",  
  height = "600px",  
  selection_env = NULL  
)
```

**Arguments**

<code>spec</code>	JSON/YAML (as R list, text, or file) or ESM JS code (text or file).
<code>specType</code>	One of "auto" (default), "json", "yaml", or "esm".
<code>data</code>	Named list of data.frames to register in DuckDB.
<code>title</code>	Optional page title
<code>width</code>	CSS or pixel width (e.g. "100%", "600px", or numeric).
<code>height</code>	CSS or pixel height.
<code>selection_env</code>	Environment to store selections in

**Value**

A Shiny application object. When the user imports a selection, the selected rows are assigned into `selection_env` as `mosaic_sel_<n>`.

# Index

`get_mosaic_selection`, [2](#), [3](#)

`list_mosaic_selections`, [2](#), [3](#)

`mosaic`, [3](#), [5](#)

`mosaicOutput`, [4](#)

`renderMosaic`, [5](#)

`runMosaicApp`, [5](#)

`runMosaicExport`, [6](#)

`runMosaicWithExport`, [2](#), [7](#)