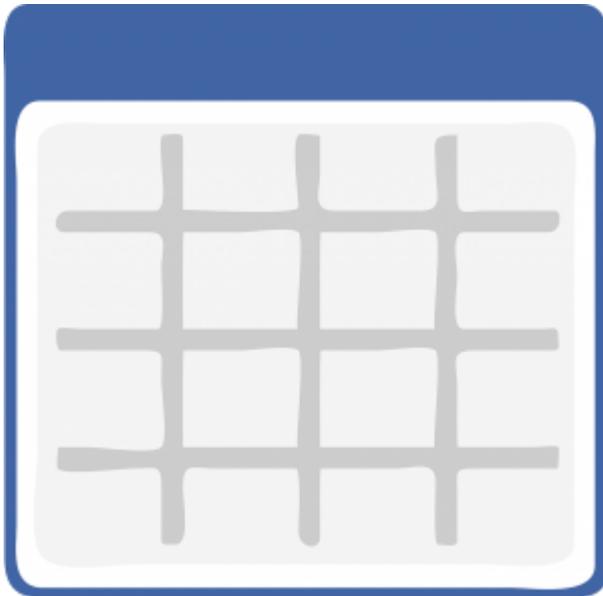


Prettify your Shiny Tables with DT: Exercises



Have you ever wanted to make your Shiny tables interactive, more functional and look better? The DT package, which stands for “DataTables”, provides an R interface to the JavaScript library “DataTables”. It allows creating high standard tables by implementing the functionalities and design features that are available through the “DataTables” library.

Even though the DT package can be used independently of Shiny, this exercise set will focus on the integration between the two packages. The exercises were built with a practical orientation in mind, and upon completing them you would be ready to tackle the vast majority of use cases for using DT with Shiny.

We will work with the light built-in dataset `datasets::orange`, that holds data about growth of orange trees. Each exercise is adding some more features/functionalities to the code of the previous exercise, so be sure to not discard the code until after you’re done with all of the exercises. Answers to the exercises are available [here](#).

Each exercises should result in a Shiny app, where the changing part is the `renderDataTable()` function.

To save time, you can use the following template, and just replace the placeholder for each exercise.

```
ui <- fluidPage(
```

```
br(), br(), br(),
fluidRow(column(width = 6, DT::dataTableOutput(outputId =
"my_datatable")))
)

server <- function(input, output, session) {
output$my_datatable <-
}

shinyApp(ui = ui, server = server)
```

Exercise 1

Generate a minimal Shiny app that displays the dataset Orange in a datatable (default values).

Exercise 2

Remove the row names.

Exercise 3

Show only 7 rows as the default display, and allow changing the number of displayed rows to either 14, 21, 28 or 35.

Exercise 4

Align the columns text (both values and headers) to the center.

Exercise 5

Remove the search box (top right) and the table information text (bottom left).



Learn more about Shiny apps in the online courses [Create Interactive Web Applications with the R Shiny Package](#) and [R Shiny Interactive Web Apps](#).

Exercise 6

Add a “copy” and “csv” buttons to allow saving the table to the clipboard and downloading it as a CSV respectively.

Exercise 7

Add a filter box for each column, at the bottom of the table.

Exercise 8

Allow selection of a single row only, rather than multiple rows (which is the default).

Exercise 9

Remove the option to sort the table.

Exercise 10

Finally, if a row was selected, display in the UI (textOutput) the selected tree and age values.

Hint: for this exercise you would have to make additional changes to the UI and server, rather than just changing the `renderDataTable()` function.