

Programming Server Logic in Shiny: Takeaways



by Dataquest Labs, Inc. - All rights reserved © 2021

Syntax

- Choose a column from a given dataset:

```
varSelectInput(  
  inputId = "example_id",  
  label = "Example Label:",  
  data = example_tibble  
)
```

- Building a checkbox input:

```
checkboxInput(inputId = "example",  
           label = "Click this if you'd like this option:",  
           value = FALSE)
```

- Building radio button inputs:

```
radioButtons(  
  inputId = "example_radio",  
  label = "Choose one of the options:",  
  choices = c("Option A", "Option B", "Option C"),  
  selected = "Option A"  
)
```

- Building a select input:

```
selectInput(  
  inputId = "example_select",  
  label = "Choose options:",  
  choices = c("Option A", "Option B", "Option C"),  
  selected = "Option A",  
  multiple = TRUE  
)
```

- Building a text input:

```
textInput(  
  inputId = "example_text",  
  label = "Write some text here:" ,  
  placeholder = "This text appears in the input, but is not actual text")
```

Concepts

- We can write Shiny apps in one of two structures:
 - a **single-file** structure, where the interface and server function are in the same file (`app.R`)

- a **multi-file** structure, where the interface and server function have their own files (`ui.R` and `server.R`)
- We can have another file called `global.R` , which can house code that we'd like to provide to both `ui.R` and `server.R` . This file is a good place to have data, helper functions, and useful variables used in the app itself. `global.R` helps separate the responsibilities of the different files.
- The `www` folder is a special directory where we can store other files that we might use in the app. This includes the actual dataset values and image files.

Further Reading

- [Shiny documentation](#)