Step 1: Identify domain logic

```
server <- function(input, output, session) {

  downloads <- reactive({
    cranlogs::cran_downloads(
      input$package,
      from = Sys.Date() - 365,
      to = Sys.Date()
    )
  })

  downloads_rolling <- reactive({
    validate(need(sum(downloads()$count) > 0, "Input a valid package name"))

    downloads() %>%
      mutate(count = zoo::rollapply(count, 7, mean, fill = "extend"))
  })

  output$plot <- renderPlot({
    ggplot(downloads_rolling(), aes(date, count)) + geom_line()
  })
}
```

Only applies to Shiny, don't export it!

Step 1: Identify domain logic

```
server <- function(input, output, session) {

  downloads <- reactive({
    cranlogs::cran_downloads(
      input$package,
      from = Sys.Date() - 365,
      to = Sys.Date()
    )
  })

  downloads_rolling <- reactive({
    validate(need(sum(downloads()$count) > 0, "Input a valid package name"))

    downloads() %>%
      mutate(count = zoo::rollapply(count, 7, mean, fill = "extend"))
  })

  output$plot <- renderPlot({
    ggplot(downloads_rolling(), aes(date, count)) + geom_line()
  })
}
```