Step 3: Generate code with expandChain()

```
> expandChain(output$plot)

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

downloads_rolling <-
  ..(downloads()) %>%
    mutate(count = zoo::rollapply(count, 7, mean, fill = "extend"))

ggplot(..(downloads_rolling()), aes(date, count)) + geom_line()
```

expandChain() returns the relevant domain logic

Step 3: Generate code with expandChain()

```
> expandChain(output$plot)

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

downloads_rolling <-
  ..(downloads()) %>%
    mutate(count = zoo::rollapply(count, 7, mean, fill = "extend"))

ggplot(..(downloads_rolling()), aes(date, count)) + geom_line()
```