

```
expandChain(output$plot(), output$summary())
```

```
# Retrieve a year's worth of daily download data
downloads <- cranlogs::cran_downloads("dplyr",
  from = Sys.Date() - 365, to = Sys.Date())
```

```
# Convert daily data to 7 day rolling average
downloads_rolling <- downloads %>%
  mutate(count = zoo::rollapply(count, 7, mean, fill =
    "extend"))
```

```
ggplot(downloads_rolling, aes(date, count)) +
  geom_line() + ggtitle("Seven day rolling average")
```

```
summary(downloads$count)
```

Using shinymeta

1. You (the app author) **identify the domain logic in your app code** so we can separate it from the reactive structure
2. Within that domain logic, you **identify references to reactive values and reactive expressions** that need to be replaced with static values and static code, respectively
3. At runtime, **choose which pieces** of domain logic to export, and in what order
4. **Present the code** to the user (in a window, as a downloadable script or report, etc.)