

Using shinymeta (recap)

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.)

Limitations and future directions

- Make `expandChain` extract `input`/reactive values as variables
- Formatting of generated code can improve
 - In particular, insignificant whitespace within source code is not preserved
- Compatibility with Shiny `async` (but should work great with both bookmarking and modules already)
- So far we've only looked at reproducing *snapshots* of app state, not necessarily “lab notebook”-style *why/how/what* over multiple iterations