U17B2R554 : Are you using the .18 debugger?
U3SJEDR96 : ouch, yeah, making `update` recurse isn't the best idea :slightly_smiling_face:
U17B2R554 : haha, right. Sometimes it can be fun to do, but it can also cause wayyy too many messages
U17B2R554 : I would be confused and then look at the debugger's cmd number increasing at an alarming rarte
U5GSY0G9J : thanks <@U3SJEDR96> <@U17B2R554> ill look into those issues
U3SJEDR96 : There aren't really any valid use-cases for doing it, though, better to split off the logic into separate functions and calling those when you need them :slightly_smiling_face:
U17B2R554 : That is true
U3SJEDR96 : (unless you're replaying history that you got from a remote source in order to restore a user session, but that's an edge-case)
U3LUC6SNS : In `Main` I have the code```
TogglePublic -&gt;
          togglePublic model
```
where `togglePublic : Model -&gt; ( Model, Cmd Msg )` with `Cmd.none` in the cmd slot of the preceding tuple.  After `togglePublic model` runs, I would like to run
```
updateCurrentDocument : Model -&gt; Document -&gt; (Model, Cmd Msg)
```
How do I do that?

U0FP80EKB : You could do something like```
let
    (updatedModel, _ ) = togglePublic model
in
   updateCurrentDocument updatedModel document
```

U0FP80EKB : Personally, I stick with the guidance of having the data structure being updated as the last parameter, so I'd change `updateCurrentDocument` to```updatedCurrentDocument : Document -&gt; Model -&gt; (Model, Cmd Msg)```

U0FP80EKB : If you want to pipeline, you could then do```
togglePublic model
|&gt; Tuple.first
|&gt; updateCurrentDocument newDocument
```

U0FP80EKB : Although I tend to go for the first version with the `let` clause to be explicit that it is ignoring the returned command
U3LUC6SNS : <@U0FP80EKB> Thanks so much! -- and also for the guidance on the position of `model`
U0FP80EKB : One thing is that you are also duplicating the knowledge that `togglePublic` returns `Cmd.none`, so I probably would do this```
let
   (updatedModel, togglePublicCmd) = togglePublic model
   (updatedWithDocument, documentUpdateCmd) = updatedModel newDocument updatedModel
in
   (updatedWithDocument, Cmd.batch [ togglePublicCmd, documentUpdateCmd ] )
```

U0FP80EKB : (fixing the names a bit)
U3LUC6SNS : What is the reason for having the data structure being updated as the last param?
U0FP80EKB : Duplicating the knowledge about the `Cmd.none` can be a pain later down the line. I'm pretty aggressive at eliminating duplication.
U0FP80EKB : It makes it better for pipelining, since `|&gt;` passes as the last parameter
U3LUC6SNS : Got it! thanks!
U0FP80EKB : So, in general, the data structure being changed can be strung through