U0LPMPL2U : All of these are different states that a call could be in
U37HUSJ4R : yes
U0LPMPL2U : I'm thinking `RecordingStatus` only makes sense for `Active` calls
U37HUSJ4R : I'd agree
U37HUSJ4R : I think that last type is pretty much spot on
U0LPMPL2U : You can think of it as "a call can be in one of N states, some of which have extra data associated (such as recording status)"
U37HUSJ4R : thank you for the help
U37HUSJ4R : great to be able to chat through these questions
U0LPMPL2U : I'm glad we found a nice solution :slightly_smiling_face:
U4RR7KX45 : how can I decode an array of custom type with json decoder?
U4RR7KX45 : ```decode List User
   |&gt; JDP.required "data" (JD.list userDecoder)
```

U4RR7KX45 : this is what I'm doing at the moment
U0LPMPL2U : You don't need the first part
U0LPMPL2U : `JDP.required "data" (JD.list userDecoder)` should work
U4RR7KX45 : it doesn't seem to work :confused: says the type annotation is incorrect. At the moment I'm using:`decodeUsers : JD.Decoder (List User)`

U0LPMPL2U : I don't think you need `JDP` here. You can say:```
decodeUsers : JD.Decoder (List User)
decodeUsers =
  JD.field "data" (JD.list userDecoder)
```

U4RR7KX45 : ah that worked
U0LPMPL2U : Json Decode Pipeline is used for decoding records with a lot of keys
U6H9E1YS3 : Hello all :slightly_smiling_face: A bit stuck on something, I have an update branch that sends a Cmd Msg with the updated model but the Cmd Msg is sending before the model has updated. What's the best way to make sure the model has updated before I send the Cmd? I've just made a variable for the list I need at the moment, but thought there had to be a better way of doing it! Any advice would be appreciated
U0LPMPL2U : I'm curious why you want to send yourself a `Cmd Msg`? Is it to prevent duplication in two branches of your `update` ?
U6H9E1YS3 : I don't understand sorry I'm not sending myself one I'm using d3 so using it to send filtered data via a port :slightly_smiling_face:
U6H9E1YS3 : I probably wasn't very clear
U0LPMPL2U : inside your `update` function, `model` always refers to the _old_ version of the model
U0LPMPL2U : All data is immutable in Elm so using the `{ model | ... }` syntax doesn't mutate `model`, but instead returns a _new_ record
U0LPMPL2U : since you want the new filtered data, you should use the same value you are assigning to the model