U6531GSTW : is there any easy way to make it work?
U6531GSTW : the different types to decode are really getting to me
U23SA861Y : so if you have a function which goes `foo : String->Time` you can use `JD.string |> JD.map foo` to get a decoder of type time
U23SA861Y : as it should be
U23SA861Y : types are there to help you
U23SA861Y : they tell you when things are incompatible
U23SA861Y : that incompatibility exists whether you explicitly call it typing or not
U0FP80EKB : Here's the decoder we wrote```
dateDecoder : Decode.Decoder (Maybe Date)
dateDecoder =
  Decode.map dateFromString Decode.string

dateFromString : String -> Maybe Date
dateFromString =
  Date.fromString >> Result.toMaybe
```

U0FP80EKB : Then used it like```
      |&gt; required "date" DateHelpers.dateDecoder
```

U23SA861Y : heh, there is a date decoder in extra
U1CE9DL9H : <@U0FP80EKB> why not let the decoder fail when no date can be parsed? so ```
decode.string |> andThen (\str ->
  case dateFromString str of
    Nothing -> Decode.fail "could not parse date"
    Just v -> Decode.succeed v
)
```
saves you from having to deal with maybes everywhere

U0FP80EKB : <@U1CE9DL9H> Not totally sure, but I think we had cases where it wasn't a failure maybe
U6531GSTW : i only partielly understand what has to be done.. is there anyway to make this code running&gt; <https://ellie-app.com/3GpdHFztdsta1/5> ? im still stuck on these errors with different types..
U0FP80EKB : oh, yeah, looking at other uses, there are cases where we want the Nothing to be there
U0FP80EKB : Since it wasn't always set
U37BS6J6N : Or should I take this to mean that my models are too complicated?
U37BS6J6N : full code listing<https://ellie-app.com/3Gs79HdmykBa1/1>

U23SA861Y : create recursive inits to help you
U23SA861Y : err, both of those are lists so yeah
U23SA861Y : ok
U23SA861Y : ahh no your model is ChatModel
U23SA861Y : yeah you should have a seperate init for each of those types and then compose them togoether
U23SA861Y : There are no `nil` types in elm
U37BS6J6N : Are you seeing a "nil"? somewhere?
U37BS6J6N : OH the MeetingInfo!
U23SA861Y : no, I mean there is not default constructur
U23SA861Y : so break it up into a few different inits
U6574P2MB : hello
U37BS6J6N : <@U23SA861Y> k thanks~ I'll give it a try
U23SA861Y : if you have things like ```
initUser: User
initUser = ...
```
it will make it easier to see where the types are breaking down

U23SA861Y : hello <@U6574P2MB>

U64FYS317 : Is there a way to define type alias B which is a superset/subset of type alias A, but without explicitly copying all the same field annotations from A?
U23SA861Y : hmm, I believe not
U23SA861Y : there is an issue for this let me dig it up
U23SA861Y : <https://github.com/elm-lang/elm-lang.org/issues/529>
U23SA861Y : you can however have a method accept a partial record
U64FYS317 : hrm. Thanks for the tip. wouldn't you just have to redefine a `PartialType`, though?
U23SA861Y : no
U23SA861Y : one sec