U3SJEDR96 : Yeah, that would work.. You would get a `I ran into a failure decoder: ` thing in the Err case, but eh
U3SJEDR96 : I suppose, if that's unacceptable, you could take that a little further and make a custom response handler - <https://ellie-app.com/3GPkxJJjHQna1/0>
U3SJEDR96 : Though you'd still have a String in the error case
U641LDZFU : Evening (or whatever time of day it is where you are) all, I am having problems with union types (again)... is it possible to have a list filled with different types and then run a filter on that list? here is a demo of what I am trying to achieve:
U641LDZFU : <https://gist.github.com/allansideas/9a2c513d1b4add499b1605abe5d69ed4>
U3SJEDR96 : Right, so you need to be explicit about the types of entries in a List. Doing that would yield something like this: <https://ellie-app.com/3GPpVLz2wdRa1/0>
U3SJEDR96 : Oh, `Record` is actually defined, missed that: <https://ellie-app.com/3GPpVLz2wdRa1/1>
U641LDZFU : heh :slightly_smiling_face:
U641LDZFU : I believe you taught me that bit last time :smile:
U5FC7BE0G : <@U3SJEDR96> Awesome! That's exactly the solution I was lookign for! String in the error case is fine. Thanks!
U641LDZFU : oh wow, so if I am getting errors on the "big array" like ```The 5th entry has this type:
{ ..., nodeContent : { ..., chatId : ..., userId : ... } }

But the 6th is:

{ ..., nodeContent : { ... } }

Hint: Problem in the `nodeContent` field. Looks like a record is missing these
fields: chatId and userId```

U641LDZFU : although the nodeContent have different type aliases
U641LDZFU : there is an error with my type aliases?
U641LDZFU : or I have to construct them in the list?
U2LAL86AY : <@U641LDZFU> in elm lists, arrays, dicts and trees and other data that resembles collections of elements - need to have the same type.
you need `chatId` and `userId` on the 6-th element - which i guess is of different type.
Can you please give your type aliases you use in that list?

U641LDZFU : Thanks, I think my problem is I need to use another union type... ```type alias ChatMessage =   { id : Int
    , chatId : Int
    , userId : Int
    , content : List (Html Msg)
    }

type alias Transition =
    { transitionType : String, toStoryNode : Int, text : Maybe String }

type alias ChatNodeContent =
    { id : Int, nodeType : String, nodeContent : ChatMessage, state : Maybe String, transitions : List Transition }

type alias NewsItem =
    { id : Int
    , content : List (Html Msg)
    , userId : 1
    , chatId : 2
    }

type alias NewsItemNodeContent =
    { id : Int, nodeType : String, nodeContent : NewsItem, state : Maybe String, transitions : List Transition }

~~~
type StoryNode
    = ChatNode ChatNodeContent
    | NewsItemNode NewsItemNodeContent```
~~~

U641LDZFU : I think I should probably rename, ChatNodeContent, and NewsItemContent, then create a new union type for the nodeContent which can be one of NewsItem or ChatMessage

U2LAL86AY : hmm.. processing .. :smile:

U2LAL86AY : if you have an image will help.

U2LAL86AY : ```type alias NewsItem =   { id : Int   , content : List (Html Msg)   , userId : 1   , chatId : 2   }
```
This should not compile

U2LAL86AY : you have values mixed with types.

U2LAL86AY : also is not so good to have `content : List (Html Msg) ` is better to just keep the data on your model , not html + data.

U641LDZFU : Ah, that was causing me trouble :slightly_smiling_face:

U2LAL86AY : i will do a rewrite here.

U641LDZFU : Ah sorry, that was an progress update to: ``` type alias ChatMessage =

~~~
    { id : Int
    , chatId : Int
    , userId : Int
    , content : List (Html Msg)
    }


type alias Transition =
    { transitionType : String, toStoryNode : Int, text : Maybe String }


type alias ChatNodeItem =
    { id : Int, nodeType : String, nodeContent : StoryNodeContent, state : Maybe String, transitions : List Transition }


type alias NewsItem =
    { id : Int
    , content : List (Html Msg)
    }


type alias NewsNodeItem =
    { id : Int, nodeType : String, nodeContent : StoryNodeContent, state : Maybe String, transitions : List Transition }


type StoryNode
    = ChatNode ChatNodeItem
    | NewsNode NewsNodeItem


type StoryNodeContent
    = NewsItemContent NewsItem
    | ChatMessageContent ChatMessage




type alias Player =
    { visitedStoryNodes : List StoryNode
    , currentStoryNode : Maybe StoryNode
    }
~~~
```

```
type alias Model =
    { player : Player
    }
```