

U0J1M0F32 : <@U1CHKCKBR> Thinking through it right now. I think I would have buttons that fire events with their corresponding key and handle it that way.
`div [onClick (KeyPressed "C")] [text "C"]` etc.

U0J1M0F32 : Yah, what <@U48AEBJQ3> suggests.

U1CHKCKBR : <@U48AEBJQ3> yeah, sort of

U1CHKCKBR : thank you, it a good starting point

U48AEBJQ3 : <@U1CHKCKBR> I got interested and tried something. It might be useful.

<<https://ellie-app.com/3GKKgFsXjbga1/1>>

U5ABF3BH7 : <@U48AEBJQ3> thanks for posting the example, it is pretty cool

U5ABF3BH7 : Hi! I have a quick question, If I have ``import FirstCall exposing (Model)`` but in my view, I want to rename Model FirstCallModel (because I already have something called Model in my view). Is there a straightforward way to do that?

U5ABF3BH7 : in other words I want to rename a type a import, is that possible?

U48AEBJQ3 : <@U5ABF3BH7> I don't think so? I think your better bet would be to not expose `Model` and instead use `FirstCall.Model` or you can `import FirstCall as FC` and then go `FC.Model`

U5ABF3BH7 : I had done that but I am getting a dependency, and I need each one for the other one. I guess I need to re-structure differently

U5ABF3BH7 : ``Your dependencies form a cycle:

???????

? Views.Pages.FirstCall

? ?

? Views.Sections.ArrangementMeeting

???????

...

U48AEBJQ3 : Yeah, dependency cycles are occasionally annoying. The easiest way is to make an extra module which doesn't depend on either. However, I find that the emergence of such a cycle is often a sign that I might want to re-think what I'm doing more fundamentally.

U48AEBJQ3 : <@U5H8JJP24> Here is another possibility: <<https://ellie-app.com/3GLHPTQgz7da1/0>>

U5H8JJP24 : <@U48AEBJQ3> Thank you so much. That looks like a really nice approach. This will take me some time for understanding it xD

U48AEBJQ3 : <@U5H8JJP24> Well, if you have any questions about it, I'm happy to answer.

U23SA861Y : mm yeah, that you might be slicing across the grain as opposed to with it.

U3SJEDR96 : `elm-community/json-extra` has an `optionalField` with those semantics, actually. No `optionalIndex` tho

U3HQQVHERX : <@U5ABF3BH7> I used to get those cycles all the time, i finally drew a directed graph of my modules on a whiteboard to help me avoid this (<https://en.wikipedia.org/wiki/Directed_graph>)

U3SJEDR96 : When you run into something like that, usually you have 2 things that are actually 3 things, or just 1.

U5ABF3BH7 : <@U3HQQVHERX> That is good idea, thanks <@U3SJEDR96> I see

U3HQQVHERX : apparently your modules should match the language/structure of the particular domain, but I've found structuring them like this is helpful: `Main.elm` (just for your `program`), `Model.elm` (for your model), `Update.elm` (for your update function), `View.elm` (for your view).

U3HQQVHERX : It depends on use case though

U3HQQVHERX : I added a small app to a server side rendered web application and I only had two modules

`Messaging.elm` (for program, update function and view) and `Data.elm` (for types, decoders, encoders, model)

U64U09S6R : hey guys, what editor do you generally use for elm? I'm using vscode currently, and found that the snippets that come with the elm extension are somewhat...illogical

U0FP80EKB : I use vim

U23SA861Y : I'm using vscode, what about it seems an issue

U64U09S6R : for example the snippet for caseof

U64U09S6R : I think I saw vscode even noting that some of the caret positions "don't seem correct" for some of the snippets, and I checked a few of them and that's indeed what it looks like

U3SJEDR96 : <<https://github.com/Krzysztof-Cieslak/vscode-elm/blob/master/snippets/elm.json>> <- I'm pretty sure they'd be open to an issue and maybe a PR for improving the snippets :slightly_smiling_face:

U3HQQVHERX : i switch between sublime text, vim, vscode and atom. lol.

U3HQQVHERX : mostly i use sublime text since that's where my motor memory is strongest.

U3HQQVHERX : <@U64U09S6R> try Atom with the elmjutsu plugin

U4JT89FGB : <@U3SJEDR96> did your demystifying decoders last night twice. big thanks! :beer:

U3SJEDR96 : twice? :smile: Good job :slightly_smiling_face:

U4JT89FGB : ha, yes twice. you put in all that work for me, i figured i'd try to get as much as i could out of it :smile:

U3SJEDR96 : That's awesome, I'm really happy to hear you found it useful! (Or _so_ mystifying you had to do it twice.
:sweat_smile:)

U4JT89FGB : no no, it actually didn't take long even the first time.. but it did help me grok a lot of what i'd been
copy-pasta for a while now

U3SJEDR96 : Cool! If you want a challenge, you could always try implementing `at` without using `at`
:slightly_smiling_face:

U5FC7BE0G : I have an API that returns either

```
...  
{ "data": {...} }  
...
```

... or in case of an error:

```
...  
{ "error": "..." }  
...
```

The status is 200 in both cases. What would be a good approach to decode this?

U1CHKCKBR : <@U48AEBJQ3> nice one, there's an odd behaviour at combo input (click and key press)

U4JT89FGB : <@U5FC7BE0G> I think `Decode.oneOf` is what you're looking for

U3SJEDR96 : <@U5FC7BE0G> What's your end-goal? Do you want to handle the `error`, do you want to get an `Err`
when you get that, ..?

U1CHKCKBR : <@U48AEBJQ3> a nice one done in vue <<http://vue-keyboard.martywallace.com/>>

U5FC7BE0G : but what would the type look like?

I know I could do something like

```
...  
type alias MyResult = { data: Maybe MyData, error: Maybe String }  
...
```

U5FC7BE0G : <@U3SJEDR96> yes an Err would be preferable

U0LPMPL2U : Could you do something like:``

```
JD.oneOf  
[ JD.field "data" myDataDecoder  
  , JD.field "error" (JD.string |> JD.andThen JD.fail)  
  ]  
...
```

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>>