

U641LDZFU : it won't have a huge impact?  
U641LDZFU : considering the model gets passed around all over the show  
U641LDZFU : (on performance)  
U48AEBJQ3 : Data rarely gets copied in Elm. Since it is immutable, you just pass or copy references.  
U641LDZFU : btw, Thanks <@U48AEBJQ3> here is a diff with your fixes, incase anyone else ever comes across this :slightly\_smiling\_face: <<https://gist.github.com/allansideas/2360e065c13a48b419fce245d0485a12/revisions>>  
U641LDZFU : Ah alright!  
U641LDZFU : Thanks again everyone for all your help!  
U641LDZFU : The door buzzer has geklingelt, so I am off for a bit :slightly\_smiling\_face:  
U62MXFL9K : ciao  
U5FC7BE0G : is there a function to create a URL query? Something like  
...  
Http.url "/baseurl" [{"somekey": "somevalue"}] == "baseurl?somekey=somevalue"  
...

Of course i could create my own but there must already be an implementation somewhere? I saw that in an earlier version this was in Http but seems to have been removed in Http 1.0.0

U5FC7BE0G : just saw that this function exists in ``lukewestby/elm-http-extra`` which does not support elm 0.18 :confused:  
U3SJEDR96 : <<https://ellie-app.com/3H9qS8T3znMa1/0>> something like that should work  
U3SJEDR96 : which also makes it easy to make an `url` function: <<https://ellie-app.com/3H9qS8T3znMa1/1>>  
U5FC7BE0G : <@U3SJEDR96> you are the best, thanks!  
U5WJZNEF5 : Hello everyone, first time I write here and as you can easily guess I'm an Elm newbie. I'm trying to write some code that takes values from a random element of a JSON array with, of course, no success. I'm able to get values from a specific element with something like `Decode.index 0 (<<http://Decode.at|Decode.at>> [ "value" ] Decode.string)` but I really don't understand how to take the element randomly.

U3SJEDR96 : Alright, so, while feasible, this is not super-trivial in Elm. Modelling randomness in an environment where every function is pure (and, as such, needs to return the same value every time the function is called) means it needs some extra stuff to pass around. Json decoders are also a somewhat involved topic, as it involves taking a blob of unknown shape with unknown types, and extracting some typed information from it.

U0FP80EKB : Personally, I would separate this idea out into two steps: decode the list, then work on randomly choosing one. Picking a random one at the point of decoding adds a potentially unnecessary complexity level.  
U0FP80EKB : Especially as <@U3SJEDR96> mentions, these two topics can be complex :slightly\_smiling\_face:  
U5WJZNEF5 : <@U3SJEDR96> yay :slightly\_smiling\_face:  
U5WJZNEF5 : <@U0FP80EKB> I thought about that too but I got stuck trying converting the JSON array to a List of I didn't know what. Dict? Records? I don't actually get the difference between the two or how I should declare the decoder in that case  
U0FP80EKB : Exactly. :slightly\_smiling\_face: Adding randomness into this is a tough added complexity.  
U0P088J1L : Hi, is there a way to alias a type name in the import statement? Something like this.  
`import SubMessages exposing (SMMsg as Msg)`

U0FP80EKB : <@U5WJZNEF5> I usually recommend people go read a lot by Brian Hicks and get his json survival guide <<https://www.brianthicks.com/json-survival-kit/>> Demystifying decoders is a great thing  
U3SJEDR96 : <@U0P088J1L> nope, though what you'll usually find people doing is calling is `Msg` within `SubMessages` and using `SubMessages.Msg` outside of that  
U3SJEDR96 : alternatively, `import SubMessages as SM` and using `SM.Msg`  
U0FP80EKB : I use the latter  
U5WJZNEF5 : <@U0FP80EKB> Thanks, I'll look into it  
U0P088J1L : OK. The reason I had to change the type name to SMMsg in SubMessages module, is because the Parent module imports ParentMessages and SubMessages, and there is an `ambiguous` naming conflict because of that.  
U0FP80EKB : His blog is really great. I learned a bunch from it, as well.  
U0P088J1L : Is it the right way to solve the issue to change one of the two Messages module's type name?  
U0P088J1L : Or am I doing something fundamentally wrong.  
U0FP80EKB : <@U0P088J1L> In this case don't bring it into your level, always fully qualify it  
U0FP80EKB : So, you'd always say `SubMessages.Msg`  
U3SJEDR96 : <@U0P088J1L> basically you could drop the `exposing`, and qualify it using `SM` or `SubMessages`  
U0FP80EKB : and the include would be `include SubMessages` without the `exposing` part (as <@U3SJEDR96> mentions)

U0FP80EKB : In general, I tend to not expose things into my module's namespace. The majority of my includes are just the module. I might alias them`include SubMessages as SM`

U0FP80EKB : It helps me keep track of where the functions are coming from, as well as keeping from name collisions.

U0P088J1L : I see. That makes a lot sense. Thanks!

U2M4VPZ9D : Hi all, I've used this <<https://github.com/halfzebra/create-elm-app>> to create an Elm app. It's very nice. I can't however work out how to turn the debugger off when serving the app with `elm-app start`. Does anyone know how to do that?

U0L27J3N0 : <@U2M4VPZ9D> why would you want to turn off the debugger?

U0L27J3N0 : I mean... it's what's designed to do... you develop your app in debug mode and when you want to switch to production, you use `elm-app build` and serve that.

U2M4VPZ9D : Sure thanks <@U0L27J3N0>

U3SJEDR96 : if you're dealing with something the debugger can't handle, you can `eject` and edit `webpack.config.dev.js` and remove the `debug: true` on line 92, tho :slightly\_smiling\_face:

U10P4FNMD : <@U2M4VPZ9D> could you please explain why you don't want the debugger in `elm-app start`?

U2M4VPZ9D : <@U10P4FNMD> I am not sure this is a problem any more. I am working on a drum machine with Elm using the Web Audio API via ports. The ports receive the audio clock on a regular basis ( at each request animation frame) so its creating a lot of debug messages. Initially I thought the sheer number of messages would interfere with the Elm scheduling, but it turns out there was a bug in my code.

U10P4FNMD : <@U2M4VPZ9D> thanks! Please feel free to hit me with a message if you'll have troubles with `create-elm-app`

U2M4VPZ9D : <@U10P4FNMD> No worries, it's actually quite cool. I can define the drum pattern in code and when you modify it and save the code, you can hear the difference. I am going to push this to git shortly.

U2SR9DL7Q : Hey, can anyone explain what this type annotation means ``matchers : Parser (Route -&gt; a) a``

U2SR9DL7Q : Not the actual types involved, but rather the structure. It's not receiving a value, but it's returning a \_Parser\_ that in itself returns \_a\_ when passed a \_Route\_ ?

U3SJEDR96 : When dealing with the url-parser library, I think it's best not to worry about the type signatures too much :x

U0D4VR7BP : the structure is `Parser x y` if that helps you making sense of the arrows

U2SR9DL7Q : oh. <@U0D4VR7BP> i see it now

U0D4VR7BP : in this case `Parser fnTakingRoute result`, it seems like

U0D4VR7BP : I don't know what it does, I'm just guessing at the meaning

U2SR9DL7Q : to the documentation I go then. I've got it all working for my own site after following the example, but this is the one part where I feel myself not understanding exactly what's going on.

U2M4VPZ9D : I pushed the code at <<https://github.com/kafecho/elm-beats>>

U3X6H8X8W : Hi everyone