

U37HUSJ4R : interesting, so in `currentTimes` how would I loop through each index?

U3SJEDR96 : you wouldn't, cause you know them - `map3 Times (index 0 int) (index 1 int) (maybe <| index 2 string)`

U0LPMPL2U : You don't need to loop. Define a decoder for a single `[855, 1900]` array, then you use `Json.Decode.list` to say you have a list of them.

U0U6ML22H : Will try to post it there, thanks!

U37HUSJ4R : I don't know the values though

U37HUSJ4R : it might change

U37HUSJ4R : for example index 2 might become []

U0LPMPL2U : `maybe <| index 2 string` gives you `Just "hello"` if there is a string in index 2 and `Nothing` otherwise

U0LPMPL2U : I think <@U3SJEDR96>'s example needs a few tweaks to handle the `[]` case

U3SJEDR96 : yeah, you'd put the whole thing in a `maybe`

U0LPMPL2U : ```startTime : Decoder (Maybe Int)

startTime =

    maybe <| index 0 int

endTime : Decoder (Maybe Int)

endTime =

    maybe <| index 1 int

message : Decoder (Maybe String)

message =

    maybe <| index 2 string

time : Decoder (Maybe Times)

time =

    map3 timeFromOptions startTime endTime message

timeFromOptions : Maybe Int -> Maybe Int -> Maybe String -> Maybe Times

timeFromOptions start end message =

    Maybe.map2 (\s e -> Times s e message) start end

times : Decoder (List (Maybe Times))

times =

    list time

...

U0LPMPL2U : something like this

U0LPMPL2U : I broke it up into really small functions to make it easier to follow

U0LPMPL2U : I find making tiny decoders and working from the bottom up is the easiest way to think through building decoders

U3SJEDR96 : Yeah, I'd have gone for `maybe <| map3 Times (index 0 int) (index 1 int) (maybe <| index 2 string)` but fair enough

U0LPMPL2U : Oh nice, I didn't think of just wrapping the whole thing with a `maybe` decoder :thumbsup:

U37HUSJ4R : nice!

U37HUSJ4R : thanks, I will take a look

U37HUSJ4R : find decoding the hardest part of elm

U0LPMPL2U : I think everyone does :slightly\_smiling\_face:

U0LPMPL2U : The trick is to start by trying to decode only the smallest piece of your data structure

U3SJEDR96 : you might be interested in going through these: <<https://github.com/zwilias/elm-demystify-decoders>>

U0LPMPL2U : Then you look into how you can combine that with other parts to decode more and more complex structures

U37HUSJ4R : will take a look

U37HUSJ4R : and good advice <@U0LPMPL2U>

U0LPMPL2U : Also, writing small decoder functions and including their type signatures helps me think through the problem

U3ZNWN526 : Okay, here's a question. Is there some reason that using sinon.js fakeTimers wouldn't work to simulate time passing with elm code?

U3ZNWN526 : I'm doing acceptance testing on my app using webdriver and trying to trigger actions that are supposed to happen every 30 seconds, and sinon.js has a fake timer mechanism that mocks all the time functions (Date.now(), trigger setTimeouts, etc.) and it's working, but Elm isn't doing anything.

U3ZNWN526 : (Does the elm runtime have it's own esoteric time mechanism or something?)

U3SJEDR96 : when you say "isn't doing anything", can you clarify what you mean?

U1L1HMOV9Q : What is the current recommended library for touch interactions?

U3SJEDR96 : I'm guessing this is related to Elm using requestAnimationFrame for rendering, with a fallback to `setTimeout(1000/60)`

U3ZNWN526 : <@U3SJEDR96> - Yeah, I mean the things that happen on a timed subscription are not happening at all (much less at an accelerated rate)

U3ZNWN526 : Ooh, so it uses requestAnimationFrame to move the runtime to the next step? Yeah, I bet that's not mocked by sinon's fake timers...

U3ZNWN526 : So if that's the case, that could explain it...

U3ZNWN526 : I wonder if there's a way to force elm to use setTimeout... ?

U3SJEDR96 : I think `window.requestAnimationFrame = undefined` should force the fallback; though all of that (sinon and disabling rAF) should probably happen before Elm is loaded

U3ZNWN526 : Yeah, that makes sense - I'll give it a shot :slightly\_smiling\_face:

U3ZNWN526 : Huh - that didn't seem to do it either, even though window.requestAnimationFrame was undefined before the app's init function was called.

U681TBBUP : <@U3ZNWN526> just dropping in but are you using a subscription to run it every 30 seconds?

U3ZNWN526 : Yeah, I sure am.

U681TBBUP : It does seem to be using `setInterval` under the hood

<<https://github.com/elm-lang/core/blob/f064b696a2b9ca93abf4c7d1e21ce9e35fbb1d0f/src/Elm/Kernel/Time.js>>

U3ZNWN526 : Well that's pretty odd! That should work with sinon's fake timers just fine...

U3ZNWN526 : But perhaps elm itself is getting confused by the fake timers, since once they are activated, time only moves forward in big jumps...

U3ZNWN526 : Well I gave up on it for now. I'll just leave that test "pending" :wink:

U3L8MM10T : For uninstalling a package, do I use `elm-package uninstall`, or do I manually edit \_elm-package.json\_ and delete appropriate line(s)? (Ditto for \_tests\elm-package.json\_) -- and then delete elm-stuff (for tests too) as well as elm.js?

U3RDE7LDP : <@U3L8MM10T> All the dependencies for your project (as of elm 0.18) are stored in the project's elm-stuff directory, so removing it from elm-package.json and deleting elm-stuff will get rid of it.

U3SJEDR96 : or removing from `elm-package.json` and re-running `elm-package install`

U4H406NET : Is elm-graphql still being maintained? In order to talk to the github graphql it needs to pass a basic auth header (note: this is in contravention of the documentation). I could fork it and do a pull request, would it be taken up?

U3SJEDR96 : <<http://package.elm-lang.org/packages/jamesmacaulay/elm-graphql/1.4.0/GraphQL-Client-Http#RequestOptions>> &lt;- shouldn't you be able to use those options? Or are you talking about a different `elm-graphql`?

U4H406NET : Different one. Sorry, didn't specify. it's base-dev/elm-graphql. I had a look at James's, I am going to talk to him about it when he resurfaces.

U3SJEDR96 : Ah, no experience with that one

U4H406NET : It seems to have made the correct design decision to start with the schema and the query, but it has not been touched for a few months.

U3SJEDR96 : (not being updated doesn't mean much when it does what it's supposed to do)

U3L8MM10T : I have function ``digitCount : Int -> IntdigitCount i =  
 abs i  
 |> toString  
 |> String.length`` and get `8` for `digitCount (1000^1000)` because that's the length of ``Infinity``. I can't imagine that this is correct behavior.

U0J1M0F32 : :thinking\_face:

U0LPMPL2U : `floor &lt;&lt; logBase 10`

U0LPMPL2U : this will give you the power of 10 of your number

U0LPMPL2U : which is the same as n-1 digits

U60SXA96 : ...and it's faster, and more accurate, and an implementation that is actually what you mean.

U0LPMPL2U : so maybe``

digitCount : Int -> Int

digitCount i =  
 abs i  
 |> logBase 10  
 |> floor  
 |> (+) 1  
 ...

U3L8MM10T : Cool. Thanks all.

U60SXF96 : Just watch out for 0.

U60SXF96 : You'll need to special case that in any case.