U3BTVEFQU : <@U2JA29EKF> yep found something similar, however how would I do another http request with this delay function?
U3SJEDR96 : So you can do `Time.sleep (100 * millisecond) |&gt; Task.andThen (makeNewRequest)`
U0LPMPL2U : You could have something like:```
subscriptions : Model -&gt; Sub Msg
subscriptions model =
  case model.state of
    Polling -&gt; Time.every second (always Poll)
    _ -&gt; Sub.none
```

U3BTVEFQU : <@U0LPMPL2U> yep, just wanted to avoid this :wink:
U0K92QFST : ilias: is it Time.sleep, or Process.sleep?
U3SJEDR96 : somewhat more complete example: `Process.sleep (100 * Time.millisecond) |&gt; Task.andThen (\_ -&gt; buildRequest |&gt; Http.toTask) |&gt; Task.attempt GotResponse`
U3SJEDR96 : Process, my bad :slightly_smiling_face:
U0K92QFST : Cool! Just making sure
U641LDZFU : So I can't still can't get unions working...
U641LDZFU : <https://gist.github.com/allansideas/60658d50f4f6b5ab9738c05f419324a8>
U641LDZFU : ```Function `unionFn` is expecting the argument to be:
MyThing

But it is:

String```

U641LDZFU : Am I missing something super obvious
U0LPMPL2U : Line 58 `text (unionFn "String")`
U0LPMPL2U : you're passing a string to `unionFn`
U0LPMPL2U : you need to give it a `MyThing`
U0LPMPL2U : in your case, `AString "String"`
U641LDZFU : ahhhhhh
U41NK9BM4 : Maybe it helps to think about `AString` as a constructor which accepts a strign as argument and return a `MyThing`
U62R599PU : For me this is just a hobby so I find it very exciting.  Luckily I started to move my JS to more of a functional approach so I at ;east have that to draw on.
U3SJEDR96 : yeah, you need to explicitly tag the values. `AString` is basically a value constructor: `AString : String -&gt; MyThing`
U62R599PU : ;east = least
U641LDZFU : so I can still do records ARecord {id = 1, thing="super"}
U641LDZFU : or do I have to do them in the Arecord 1 "super" style?
U641LDZFU : (I will try now)
U0LPMPL2U : If you're adding onto the `MyThing` type you'll want `ARecord {id = 1, thing="super"}`
U41NK9BM4 : Same here, at least for now I'm writing Elm code for fun
U0LPMPL2U : `Arecord 1 "super` can be used with a _type alias_ like `type alias Arecord = { id : Int, thing : String }`
U0LPMPL2U : <http://faq.elm-community.org/#what-is-the-difference-between-type-and-type-alias>
U641LDZFU : ah perfect!
U0LPMPL2U : with the _type_, `ARecord` is wrapping the record. with the _type alias_ `Arecord` is the record.
U641LDZFU : thank you so much!
U641LDZFU : got it :slightly_smiling_face:
U5DJNFLUD : Opinion time: If you get to the point where you need to split a data model off from `Main`, do you also tend to create an `update` function on your new Model and call that from `Main`?
U23SA861Y : after a few thousand lines of elm, I haven't got that far.
U1P6FFJ64 : <@U5DJNFLUD> yes. You can have a look at this post, with a nice example of a large application to see how to split your code: <https://dev.to/rtfeldman/tour-of-an-open-source-elm-spa>
U23SA861Y : I got into alot of trouble thinking that I need new files after a hundred lines or so because of the stigma associated with OO programming
U1P6FFJ64 : About files life and how do they grow, Evan did a nice talk at Elm Europe about this topic, but the video is not published yet. It will be published at some point in this youtube channel: <https://www.youtube.com/channel/UCT5HLUjjXdqUSUnpblFNOwQ/videos>
```

U5DJNFLUD : Yeah! I've seen the talk
U5DJNFLUD : I like the idea of doing that page-wise
U5DJNFLUD : Pretty nice separation of concerns there
U3BTVEFQU : <@U3SJEDR96> (cc <@U0LPMPL2U> <@U2JA29EKF>) I wasn't able to use your example :confused: How would you proceed with code like here: <https://ellie-app.com/3GkfJWm6PM9a1/0>
U0J1M0F32 : I still try to keep most everything on `Main` unless its very page specific and has the possibility of either growing exponentially or messing with data from another "page" in either naming convention or type.
U23SA861Y : uh oh, seems like ellie is having issues
U3BTVEFQU : it wasn't me!! :wink:
U2JA29EKF : Sorry, won't be able to look tonight.
U3SJEDR96 : ```sendDelayedRequest : (WebData String -&gt; msg) -&gt; Cmd msg
sendDelayedRequest tagger =
    Process.sleep (200 * Time.millisecond)
        |&gt; Task.andThen (_ -&gt; Http.toTask request)
        |&gt; RemoteData.fromTask
        |&gt; Task.perform tagger
~~~

U48AEBJQ3 : ream88: <https://www.youtube.com/watch?v=NbosplYmAT0>
U23SA861Y : umm, request isn't defined....
U3BTVEFQU : <@U3SJEDR96> ok this does work. However I still don't get it.. do you know any good introduction into cmds vs tasks?
U0F7JPK36 : jonf: something weird happened with aws. looks okay now
U3SJEDR96 : In <@U3BTVEFQU>'s code, it is :wink:
U3BTVEFQU : Btw <@U3SJEDR96> thx!!! :slightly_smiling_face:
U3SJEDR96 : Though this probably makes more sense:
U3SJEDR96 : ```sendDelayedRequest : Time.Time -> Http.Request a -> (WebData a -> msg) -> Cmd msg
sendDelayedRequest delay request tagger =
    Process.sleep delay
        |> Task.andThen (_ -> Http.toTask request)
        |> RemoteData.fromTask
        |> Task.perform tagger
```

U3BTVEFQU : <@U2JA29EKF> no problemo, <@U3SJEDR96> fixed it for me.. However I'm still on the task to understand it properly :wink:
~~~