

U2J1FUQTZ : Anyone could be so kind to post an example of code showing how to use the function andThen of elm-tools/parser ?

U3SJEDR96 : Hm, it's hard to find a self-contained example

U3SJEDR96 : I've used it for recursive stuff where I build up a list of intermediary terms, though I think there's an example of that in the docs, too

U3SJEDR96 : <<https://github.com/elm-tools/parser/tree/2.0.1#delayed-commits>> specifically, in that part

U3SJEDR96 : it's not that different from `Json.Decode.andThen`, though the use-cases tend to be a little different

U3SJEDR96 : Is there a specific reason for why you need help with `andThen`?

U3SJEDR96 : It might be easier to walk you through your use-case and hope it "clicks".. :slightly_smiling_face:

U2J1FUQTZ : The terms between commas in the case I shown before could contain also "(" and ")" I'm trying to change the first and the last parenthesis to something else before using the parser

U2J1FUQTZ : the pattern is; head(<everycharexceptcomma>;...)

U4872964V : i'm going to look at that sortable library you tried to implement, perhaps we can make a super nice version in Elm

U2W28MK6K : Hi all! I've been struggling with being able to send a request that returns a 204 No Content. I've explored creating a low level request like so:

```

...
myRequest msg payload =
  let
    req =
      Http.request
        { method = "POST"
        , headers = []
        , url = "/my-url"
        , body =
            (Http.jsonBody &lt;|
              Encode.object
                [ ( "field1", Encode.string &lt;| payload.field1 )
                ]
            )
        , expect = Http.expectStringResponse (\response -&gt; Ok response.body)
        , timeout = Nothing
        , withCredentials = False
        }
  in
    Http.send msg req
...

```

U2W28MK6K : i've also tried:

```

...
expect = Http.expectString
...

```

to no avail

and the result that I always get is an Error regarding not being able to parse invalid JSON

U68RLL9FF : Hey guys, wondering im struggling with types annotation where the type is two seperate words, but i think it clicked...if i get this right...when we see `view: Model -> Html String` its basically shortHand for `view: Model -> { recordTypeA : Html, recordTypeB: String }`

U68RLL9FF : If i'm wrong, i'd love to understand what is really happening

U1UGYHGCA : I think you're wrong

U1UGYHGCA : The raw type is `Html a` where `a` can be anything at first. You will then say that, in your particular use case, it will be a `String` becoming a `Html String`

U1UGYHGCA : For example, you have `Maybe a` using the same pattern where `a` will be the type of the value inside the `Maybe`

U1UGYHGCA : If you have an `Int` inside it, it will be a `Maybe Int`

U1UGYHGCA : For `Html`, the generic type `a` is the type of msg that the html can trigger

U68RLL9FF : Thanks <@U1UGYHGCA> this makes much more sense than my initial though

U6CCW4KV1 : I already posted this in general, but I guess it's actually a newbie question :sweat_smile: