

U153UK3FA : The many ways of breaking up your app become obvious once you understand enough of the basics

U2D7NUGS1 : <@U3SJEDR96> Thanks again for your tremendous help. Here's my yield of today:

<<https://github.com/lzrski/elm-polymer-scaffolder>>

U2D7NUGS1 : As mentioned before the goal is to reimplement `scripts/scaffold` from `elm-polymer` in Elm. Sorry for lack of README or any other docs. I'm in a hurry.

U2D7NUGS1 : Good night to you and all other mighty folks of Elm :waning\_crescent\_moon:

U5D4VHEN7 : I am trying to write a JSON decoder for a record that has a field `status` who's type is `Status` where the possible values are `Past, Present, Future, None`. The JSON response with the status data will be structures like `{ "status" : "past", ...other\_fields }`. I am confused as to how I go about decoding `past` to `Past`. So far I have``

dayBreakdownDecoder : Decoder DayBreakdown

dayBreakdownDecoder =

decode DayBreakdown

|&gt; required "grade" dayGradeDecoder

|&gt; required "status" dayStatusDecoder

dayStatusDecoder : Decoder DayStatus

dayStatusDecoder =

Decode.string `andThen` decodeDayStatus

decodeDayStatus : String -&gt; DayStatus

decodeDayStatus dayStatus =

case dayStatus of

"past" -&gt; Past

"present" -&gt; Present

"future" -&gt; Future

other -&gt; None

...

which is obviously wrong. Anyone able to point me in the right direction?

U0CLDU8UB : That's not obviously wrong in my opinion

U0CLDU8UB : The only problem is the "error" case, where `status` is none of those strings.

U0CLDU8UB : You'll have to decide if you want that to be a success or a failure

U5D4VHEN7 : Thanks for the reply <@U0CLDU8UB>. I apologize, because I have actually just left that part out. I'll edit it

U0CLDU8UB : Okay. That to me seems like it should work. What is your error message?

U5D4VHEN7 : And from this I get:``

I ran into something unexpected when parsing your code!

206| Decode.string `andThen` decodeDayGrade

^

I am looking for one of the following things:

end of input

whitespace

Detected errors in 1 module.

...

U0CLDU8UB : Oh right, the backtick syntax of course

U0CLDU8UB : That was removed in 0.18

U5D4VHEN7 : ahhhhhh, got it. Let me try it without. Is there an alternative, or must I just place the args in order?

U0CLDU8UB : So now it's``

dayStatusDecoder : Decoder DayStatus

dayStatusDecoder =

Decode.string

|&gt; andThen decodeDayStatus

...

U5D4VHEN7 : !!! Works! Thanks so much <@U0CLDU8UB> :slightly\_smiling\_face:  
U0CLDU8UB : Awesome! Glad I could help!  
U5D4VHEN7 : Actually, was still doing something wrong, but fixed it. :slightly\_smiling\_face: Here is the final solution  
(with poor naming prior to refactoring):``

```

dayBreakdownDecoder : Decoder DayBreakdown
dayBreakdownDecoder =
  decode DayBreakdown
  |&gt; required "grade" dayGradeDecoder
  |&gt; required "status" dayStatusDecoder

dayStatusDecoder : Decoder DayStatus
dayStatusDecoder =
  Decode.string
  |&gt; Decode.andThen doDecodeDayStatus

doDecodeDayStatus : String -&gt; Decoder DayStatus
doDecodeDayStatus dayStatus = Decode.succeed (decodeDayStatus dayStatus)

decodeDayStatus : String -&gt; DayStatus
decodeDayStatus dayStatus =
  case dayStatus of
    "past" -&gt; Past
    "present" -&gt; Present
    "future" -&gt; Future
    _ -&gt; NoStatus
``

```

U6303RTK7 : strange issue  
U6303RTK7 : I'm seeing this error: ``duration : Span -&gt; Int  
duration span = span.duration  
``  
``  
`span` does not have a field named `duration`  
``

```

U6303RTK7 : ``type Span
= Span { id : Int, duration : Int }
``

```

U5D4VHEN7 : I could be wrong, but I think you need to use a type alias instead?``

```

type alias Span =
  { id: Int, duration : Int }
``

```

U5D4VHEN7 : I very well could be wrong  
U6303RTK7 : that seems to have resolved the issue, thanks :slightly\_smiling\_face: