U0JFGGZS6 : I mean, yes, I could use that too...
U1CE9DL9H : seems that there is <https://github.com/elm-lang/elm-compiler/commit/5fe9bb9602163927295a6b37a64fd47f982d502a>
U0H3A9XLN : Well, lets be honest - "thinking" part is kinda technology independent. If you will do something first in Elm and then reimplement in in JavaScript it will probably take less time as well.
U0H3A9XLN : I do not compare languages here, just trying to say that this fact does not prove any point
U2GTQM83A : <@U0H3A9XLN> Yep that did contribute.
U0H3A9XLN : And I am wondering when people start comparing Elm to TypeScript for example. Of course comparison to JavaScript gives more advantages, but more and more people are using typed supersets nowadays.
U0H3A9XLN : For example while doing TypeScript for almost a year I haven't had any single runtime exception :)
U0H3A9XLN : I still like Elm more. And I believe taking higher targets for comparison gives a good motivation for moving forward :)
U0CL0AS3V : :heart: thanks!
U2SR9DL7Q : Can anyone help with a bit of restructuring advice? Splitting my project into submodules currently. I'm delegating my main views to those of the submodule:```
viewContent : Model -> Html Msg
viewContent model =
    case model.route of
        ...

        PortfolioRoute ->
            Dominoes.view model.dominoes
```
There's a naming conflict where the view functions in _Dominoes.elm_ return type `Html Msg` but Dominoes.elm has it's own Msg type defined. I'm thinking I should change the name of that message type to like, `DominoesMsg` or something, but I think my bigger confusion is the whole use of `Msg` vs `msg` which I thought I understood, but perhaps I don't.

U2GTQM83A : Hey <@U0CL0AS3V> and <@U23SA861Y> , I did it! Solved the puzzle.
This is the requirement: I need to make sure that if I'm decoding something, my request must ask for it to be embedded, tying the embedding on the decoder type.
For that we would need:
- Have the API call be something like this `apiRequest : Decoder a -&gt; Embedding a -&gt; Cmd (Webdata a)`
- Have a `Embedding a -&gt; List String` function that would be polymophic enough to take any `a` and return an appropriate list of strings
- Be able to compose our embeddings so that we could have a function like this `func : Decoder author -&gt; Embedding (Book author)`

We faced the following problems:
- We can't create a function that will accept a class of allowed types to transform them to a list of strings
- We do not have a value of the type we are embedding at the time of the request.

Now to the code!

Here is our magic type

```
type Embedding a
    = NoEmbedding (Maybe a)
    | Embedding (List String)
```

With this type I can create an `Embedding Anything` that doesn't need to contain `Anything` at all.
It can just contain a list with the fields that must be embedded for that type.
Or, if it is a string, it can say that there is no Embedding. And it can create an `Embedding Anything` by saying `NoEmbedding Nothing`.

You can see a compiling and working example here: <https://ellie-app.com/3RJBsCTwWgVa1/0>

U2GTQM83A : With this I was able to write our desired api call function like this:```
```

```
someApiRequest : Decoder a -&gt; Embedding a -&gt; Http.Request a
someApiRequest decoder embedding =
    let
        endpoint =
            "/books/?embed=" ++ (List.foldr (++) "" &lt;| embeddingToList embedding)
    in
        Http.get endpoint decoder
```

U2GTQM83A : The drawback is that because there is no type relation between `Embedding a` and how `a` actually looks like, `Embedding Author` could look like anything and be defined many times in different ways. To avoid that I will have to make sure to only define one function to generate each `Embedding`.
U0JFGGZS6 : Yes. If you have a top-level `DominoesMsg Dominoes.Msg`, then you can do this:
U0JFGGZS6 : `Dominoes.view model.dominoes |> Html.map DominoesMsg`
U2SR9DL7Q : Html has a map...
U0JFGGZS6 : alternately, you can do the mapping in `Dominoes.view` by passing in the mapping:`Dominoes.view (\submsg -> DominoesMsg submsg) model.dominoes`

U2SR9DL7Q : Sorry, taking time to respond, but I need to make sure I understand. So I can define `DominoesMsg` in the top level of my Main as The Msg type in dominoes, and pass it to my dominoes view
U0JFGGZS6 : but that's often used in 'reusable' sub-views which have generic `Html msg` return types, and are intended to be called from different 'parent' call sites.
U2SR9DL7Q : but i think my issue is that the compiler is saying that my main view function is emitting a _Html Msg_ in one and a _Html Dominoes.Msg_ in another
U0JFGGZS6 : that's what Html.map helps with. It wraps `Dominoes.Msg` in `Msg`.