

U3LUC6SNS : <@U3SJEDR96>, thanks again! Got it working like I want with `identity`
 U663M2MB7 : Okay, thank you guys. So nesting is not a recommended representation of data in Elm?
 U0CLDU8UB : Nested records are fine as long as the data isn't updated often.
 U0CLDU8UB : When you do want to update a nested record, it's nice to make that into a function: `{ model | favoriteCat = updateCat "meow" model.favoriteCat }`
 U663M2MB7 : Right, this is data that is updated pretty much constantly though.
 U663M2MB7 : But only certain parts of it, so I guess a function that updates the stuff I want to update, makes sense
 U604S603Y : are different cases of a discriminated union interchangeable, like base class and derived class in OOP?
 U604S603Y : I'm trying to have only one field in the top-level model of my SPA which then "contains" the page data of the currently active page. So I tried making a DU with all the different page data types I have, one case for every page. But then I have problems calling the different pages' update functions, because they expect the correct data and not the top-level DU
 U604S603Y : Then I tried pattern matching on the DU and only calling the update function when the case is the right one - but that results in problems with the overall return value of the top-level update function branch
 U3SJEDR96 : you'd need to "rewrap" that return value with the correct tag
 U3SJEDR96 : ```LoginModel model ->
 (Login.update model msg) |> Tuple.mapFirst LoginModel |> Tuple.mapSecond (Cmd.map LoginMsg)
 ```

U3SJEDR96 : or something along those lines  
 U604S603Y : now there's only one error left. The code is:

```
``` SchnellcheckMsgs msg -&gt;
  let
    ( schnellcheckViewModel, cmd ) =
      case model.subpageData of
        SchnellcheckModel sm -&gt;
          (Schnellcheck.Update.update msg sm)
            |&gt; Tuple.mapFirst SchnellcheckModel
            |&gt; Tuple.mapSecond (Cmd.map SchnellcheckMsgs)

        _ -&gt;
          ( None, Cmd.map SchnellcheckMsgs Schnellcheck.Types.NoOp )
  in
    ( { model | subpageData = schnellcheckViewModel }
      , Cmd.map SchnellcheckMsgs cmd
    )```
```

where `None` is a case of the DU stored in `model.subpageData` and `NoOp` is a case of the same Message DU stored in `SchnellcheckMsgs`.

The error is in the `(None, Cmd.map SchnellcheckMsgs Schnellcheck.Types.NoOp)` line:

```The 2nd argument to function `map` is causing a mismatch. - Function `map` is expecting the 2nd argument to be:

Cmd Schnellcheck.Types.SchnellcheckMsg

But it is:

Schnellcheck.Types.SchnellcheckMsg```

U604S603Y : the code resides in the top-level update function in my Main.elm

U3SJEDR96 : How about `Cmd.none` instead? :slightly\_smiling\_face:

U604S603Y : BLACK MAGIC

U604S603Y : it's working now! thanks <@U3SJEDR96>

U62PV9CPN : in elm-html is there a way to get the `&lt;em&gt;` tag to evaluate in this example:

```

```
span [] [ text "Some &lt;em&gt;emphasized&lt;/em&gt; text" ]
```

```

U3SJEDR96 : The proper way would be `span [] [ em [] [ text "emphasized" ] ]`

U3SJEDR96 : 2 workarounds exists; using elm-markdown on the string or using an unsafe hack

U180KMGRE : ```span

[]

```
[text "some "
 , em [] [text "emphasized"]
 , text " text"
]
```

U180KMGRE : unsafe hack way: ``

```
span [property "innerHTML" (string "Some emphasized text")] []
```