

U4872964V : But your usage of extensible records is not wrong as a general principle for grouping fields

U236M9FH9 : That way I don't need ```

```
let customer = model.customer
```

```
  updatedCustomer = { customer | name = newName }
```

```
in { model | customer = updatedCustomer }
```

```
```
```

in every case branch

U236M9FH9 : I usually just use extensible records to limit the fields my functions expect/operate on, instead of using them to define actual data

U5M297AG7 : Yes, I am updating `Customer` all at once. The extra field, message, that will exist in the UI's model was an initial concern, as I do not wish to communicate that as part of the `Customer` resource; the code serializing the `Customer` resource only serializes the fields that make up a customer, however I am thinking that was your concern with this approach, <@U4872964V>?

U4872964V : Well, I'm not really concerned :) getting the model right is important but refactoring is easy

U5M297AG7 : Ah <@U236M9FH9>, I had not considered that! I believe that would ameliorate the pain of updating nested records

U5WD40ZA9 : Seems to work, not the easiest thing I've ever set up though.

U5M297AG7 : Thank you <@U236M9FH9>, that approach worked out nicely

U236M9FH9 : Yup :slightly\_smiling\_face:

U37HUSJ4R : Hi everyone

U37HUSJ4R : If my model looks like:```type alias Application =

```
{ id : Int
 , term : Int
 , amount : Int
}
```

type alias Model =

```
{ application : Application
}
```

```
```
```

and I am trying to update the `term` value, I have `onInput UpdateTerm` on an input inside my update case statement how do I update this value?

U37HUSJ4R : so far I have `UpdateTerm term ->`

U37HUSJ4R : but unsure how I can update only the `term` value

U0EUHKVGB : <https://guide.elm-lang.org/architecture/user_input/text_fields.html>

U37HUSJ4R : Yes I've seen this

U37HUSJ4R : my issue is more updating the field inside application

U3SJEDR96 : you'd do it in two steps, something like this:```

```
let
```

```
  application = model.application
```

```
  updatedApplication = { application | term = updatedTerm }
```

```
in
```

```
  ({ model | application = updatedApplication }, Cmd.none)
```

```
```
```

U0LPMPL2U : <<http://faq.elm-community.org/#how-can-i-change-the-value-of-a-nested-field>> ?

U3SJEDR96 : or, yeah, that :smile:

U0LPMPL2U : Alternatively, you break it into two functions:

```
```
```

setTermOnApplication : Int -> Application -> Application

setTermOnApplication term app =

```
{ app | term = term }
```

setTermOnModel : Int -> Model -> Model

setTermOnModel term model =

```
{ model | application = setTermOnApplication term model.application }  
...
```

U37HUSJ4R : cheers everyone :smile:

U1EEBCQM6 : Hi guys & gals

U1EEBCQM6 : lol

U1EEBCQM6 : So slackbot needs to get smarter

U1EEBCQM6 : hahaha

U1EEBCQM6 : question... Is it possible to decode a deeply nested array into a flat `List` and not a nested type?