U2SR9DL7Q : I also use mdl, and also have a types module, with a similar set up. But I just use import Types exposing (..) and it all works <@U68N7EGVB>

U0LPMPL2U : You want to `module Types exposing (Model, Msg(..))`

U0LPMPL2U : otherwise only the `Msg` type gets exposed but not the constructor `Mdl`

U68N7EGVB : Hm. That's what I'm trying to do, but it seems like there are always those three errors: 2x `Cannot find variable Mdl` and once `Cannot find pattern Mdl`.
Importing `Model` works perfectly fine so the module is seen.

U68N7EGVB : So it looks like it's just importing `Msg` and not `Mdl`, just like you are saying.

U57KYFW67 : ah yeah

U68N7EGVB : Ok, reading is hard, it's time to go to sleep. I was doing `import Types exposing (Msg(..))` but not `module Types exposing (Msg(..))` as <@U0LPMPL2U> said. So I was trying to import everything while not exporting everything.
My bad, thanks for the help everyone!

U2GPAEU1L : Hey everyone, I see a lot of people trip-up / not properly understand extensible records, I made a tutorial explaining them and also explaining the common pitfalls, if you wanna learn about extensible records I recommend spending 5-10 minutes of your time going through this tutorial:
<http://codetidbit.com/#view/snipbit/596d2ebf7890d775c05191f7/introduction>

Cheers :smile:

U2GPAEU1L : (friendly for beginners - experts, anyone who doesn't understand extensible records)

U5ABF3BH7 : <@U2GPAEU1L> Thanks!

U5ABF3BH7 : Hi everyone, I am getting a mismatch error and I haven't been able to fix it! The error says I am missing one argument but I don't get it. I am trying to wrap the address fields into an address model as it needs to be reusable. Could someone help me to figure out how to fix it? I have tried many things but it is not working.
Here is my error
```
-- TYPE MISMATCH ------------------------------- ././Views/Sections/Location.elm

The argument to function `AddressMsg` is causing a mismatch.

271|                                                            AddressMsg Address.ChgCountry)
                                                                   ^^^^^^^^^^^^^^^^^^^^^^^
Function `AddressMsg` is expecting the argument to be:

    Address.Msg

But it is:

    String -&gt; Address.Msg

Hint: It looks like a function needs 1 more argument.

Detected errors in 1 module.
```

I have a Location.Model where the address is a submodel
 `` type alias Model =
    { type_ : String
     address : Address.Model
 }``

in my Address module, Address has fields ;
```
 Address.Model =
    { id : Maybe Int
    , address1 : String
    , address2 : String
```

    , address3 : String
    , address4 : String
    , country : String
 }
```

 Address.Msg =
    ChgAddress1 String
  | ChgAddress2 String
  | ChgAddress3 String
  | ChgAddress4 String
  | ChgCountry String
}

In my view, I have a config that includes the Model (Location.Model)
```
type alias Config msg =
    { model : Model
    , toMsg : (String -&gt; Msg) -&gt; String -&gt; msg
    , toClickMsg : Msg -&gt; msg
    }
```

`` div [ class "col-xs-7" ]
    [ select [ class "select form-control", onInput (config.toMsg (AddressMsg Address.ChgCountry)), value config.model.address.country ]
        AddressHelper.countryOptions
    ]

```

```
| AddressMsg Address.Msg

```
AddressMsg submsg -&gt;
     let
       updatedAddress = Address.update model.address submsg
     in
    { model | address = updatedAddress}
```
Sorry for the long message, I am trying to give the context. So my question is, how can I wrap the AddressMsg Address.ChgCountry so that I don't get the error?

U3HQVHERX : `Address.ChgCountry` is expecting a string because your Address modules Msg is: ` | ChgCountry String`
U5ABF3BH7 : My location msg used to have the field country, and the message ``onInput (config.toMsg ChgCountry)`` worked fine. Now that I am trying to wrap it, I get the error mentioned
U5ABF3BH7 : sorry not msg but model
U3HQVHERX : what is `submsg`?
U5ABF3BH7 : I think submsg is (ChgCountry str)
U5ABF3BH7 : <@U3HQVHERX> The following code provides a string already
U5ABF3BH7 : ```div [ class "col-xs-7" ]   [ select [ class "select form-control", onInput (config.toMsg (AddressMsg Address.ChgCountry)), value config.model.address.country ]
       AddressHelper.countryOptions
   ]```

U5ABF3BH7 : since it is a select
U3HQVHERX : what about `onInput (config.toMsg (AddressMsg Address.ChgCountry))` --&gt; `onInput
```

Address.ChgCountry` and change type to `Html msg`
U57KYFW67 : Ack. What is the command to genreate a new elm-package.json file?
U5ABF3BH7 : It doesn't work either, it needs to be wrapped.
U3HQVHERX : <@U57KYFW67> `elm-repl` will make one incidentally i think
U57KYFW67 : oh, I had an elm-stuff directory!
U3HQVHERX : i'll test this out and see what I find :slightly_smiling_face:
U3HQVHERX : <https://gist.github.com/nqthqn/5843404710c279427203a582a1bef488>
U3HQVHERX : So if you have a module `Foo` that exposes `Msg` and `update` you can do this in the place you are using the module

```
type Msg
    = Bar
    | FooMsg Foo.Msg
```

then pattern match just as you are doing.

And in the view:

`button [ onClick (FooMsg Foo.Uno) ] [ text "Foo.Uno" ]`