U5US3G22C : I wanted to decode that from a request I got. So I first had to figure out how to setup the get request, then what decoders I need (ok lets just use `value` and handle it down the line). Now how do I check the "type" field and decode based on what it is
U5US3G22C : I've got it figured out now, but it took me a while and doing so in the original project (as I just used my workflow i described) would have led to a bunch of errors that would have kinda put me off
U5ABF3BH7 : Here is what I'm seeing. I have a Country field created using [this line of code, nested in a condition].
U5ABF3BH7 : ```, div [ class "col-xs-7" ]                              [ select [ class "select form-control", onInput (config.toMsg Location.ChgCountry), value config.model.country ]
                                Addresses.countryOptions
                             ]```

U5ABF3BH7 : When a country other than United States is selected, the UI displays a different set of fields above the country field. This works as expected. However, what I'm finding, is that when I select a different country than US, "United States" continue to display instead of the country selected. I have to once again select a country for the field to update.

If the line creating the country field is in the main view, it works as expected, if it is in the subview, displayed on condition of the m model, then I have that issue.

U48AEBJQ3 : <@U5US3G22C> Understanding composible data structures is *the* biggest stumbling block for new Elm users despite many difficulties. It's a big problem for a lot of FP languages.
U5ABF3BH7 : <https://screencast.com/t/GnvOvy2SMx1>
U48AEBJQ3 : <@U5ABF3BH7> It *seems* to be a problem with the virtual dom losing track of what goes where. Have you tried setting `selected` on the selected option?
U48AEBJQ3 : You might also try <http://package.elm-lang.org/packages/elm-lang/html/2.0.0/Html-Keyed#node>
U5ABF3BH7 : I tried that selected model.country, but get the error that selected is expecting a Bool
U5ABF3BH7 : ```The argument to function `selected` is causing a mismatch.
88|                                                                                    selected model.country
                                                                                            ^^^^^^^^^^^^^^

Function `selected` is expecting the argument to be:

    Bool

But it is:

    String```

U5ABF3BH7 : Since I don't know in advance what the bool is I am a bit confused as what I should put for selected
U48AEBJQ3 : You have to put `selected True` in the currently selected `option` tag. But I think you should try `Keyed.node` first.
U5ABF3BH7 : Ok, thanks, I will try that
U48AEBJQ3 : ```countryOptionList id =
    List.map (\country -&gt; option [selected (country.id == id), value country.id] [text country.name])
```

U5ABF3BH7 : Ok, thanks <@U48AEBJQ3>
U5HM74BD0 : Is there an Elm package for encoding my model to json? I want to store my model in localStorage.
U48AEBJQ3 : <@U5HM74BD0> `Json.Encode`?
U5HM74BD0 : <@U48AEBJQ3>, how would you work with tuples?
U48AEBJQ3 : Either in separate fields in an object or in a list.
U48AEBJQ3 : ```tupleToList : (a -&gt; Value) -&gt; (b -&gt; Value) -&gt; (a, b) -&gt; Value
tupleToList xEncoder yEncoder (x, y) =
    Encode.list [xEncoder x, yEncoder y]
```

U5HM74BD0 : <@U48AEBJQ3> I see, yes.  Okay, I'll give it a whirl.  Am I thinking about this right: the model gets encoded into json; that json gets passed to javascript side, which can then convert it to a string representation; it can then store that string in localStorage. To decode, reverse the process?
U48AEBJQ3 : A `Value` isn't actually JSON, it is a JavaScript value. You have to call <http://package.elm-lang.org/packages/elm-lang/core/5.1.1/Json-Encode#encode> to turn it into a JSON string. You

are free to do this on either side of the port.
U5HM74BD0 : Okay, thanks very much.