U0JBSEGHY : At the bottom of the docs there is `listDefault` and `list` which gets me close to what I want. `listDefault` shows all the `metas` but I lost the name of the person. So I would like to somehow map over each person and count their `meta` lists.
U27KX9MEJ : <@U3SJEDR96> Ok, but what if we suppose that content-records of my union type always have the same loadingState? (but ofcourse not exactly the same). Is there any way to pattern match this?
U3SJEDR96 : No, but they you'd be able to represent the whole thing differently - `type Widget = Widget { type_ : widgetType, loading : Bool, ... }`
U0JBSEGHY : I was actually thinking like a `map` `filter` `listDefault` or something along those lines would get me the `metas` individually
U0JBSEGHY : So I get the `keys` to the `Dict` then I do a `map` using the keys. Filter the `Dict` then use the listDefault to extract the metas.
U0JBSEGHY : Actually that still wouldnt' get me it lol
U27KX9MEJ : <@U3SJEDR96> thank you for help, but as far as i understand we can't use type as variable like this?
```
test : WidgetState -&gt; WidgetState
test widgetState =
    case widgetState of
        widgetStateWithLoadingState { rec | loadingState : Bool} -&gt;
            widgetStateWithLoadingState { rec | loadingState = True }
        _ -&gt;
            widgetState
```

U3SJEDR96 : Correct - trying to that _usually_ means your data can be structured differently
U27KX9MEJ : how can i share with you my structure for better understanding?
U27KX9MEJ : what do you need to know about it?
```
type WidgetState
    = TabsState TabsStateRecord
    | SidebarState SidebarStateRecord
    | ComboboxState ComboboxStateRecord
    | FormState FormStateRecord
    | TableState TableStateRecord
    | FileManagerState FileManagerStateRecord
    | ListState ListStateRecord
   ...
```

U27KX9MEJ :
```
type alias TabsStateRecord =
    { active : String
    }
type alias TableStateRecord =
    { selector : Selector
    , options : Bool
    , minimizedGroups : List String
    , modals : Dict String ModalBoxRecord
    , selected : List String
    , loadingState : LoadingState
    }
```
That's example of state record with loadingState and witout it.

U0JBSEGHY : <@U3SJEDR96> doesn't a map just transform what it maps over though?
U0JBSEGHY : I really want to map over the Dict and cause a side effect
U3SJEDR96 : That's a pretty complex structure. One option is to create a `type Widget = Instant InstantWidgetState | Deferred Bool DeferredWidgetState` where those two `WidgetState` types are sub-sets of your current `WidgetState`. Then again, is it really that bad to define this one function? It seems like you're using pretty heavy encapsulation, and that leads to more code to reach into separate cases.
U0JBSEGHY : The dot notation doesn't seem to help me either
U3SJEDR96 : <@U0JBSEGHY> - What side-effect are you trying to cause? I'm not sure I'm following?
U0JBSEGHY : Given this ```Dict.fromList [("Jazz",{ metas = [{ phx_ref = "/cEkW0uji8g=", payload = { online_at = "1500725263", device = "browser" } }] }),("Wade",{ metas = [{ phx_ref = "mIHOmevHztc=", payload = { online_at =

"1500725467", device = "browser" } },{ phx_ref = "pxEwxgqeucY=", payload = { online_at = "1500725252", device = "browser" } }] })]```
U0JBSEGHY : I want to be able to count how many sessions each user has and show them.
U0JBSEGHY : `listDefault` in the package provides me with this when I provide the `Dict` ```[{ metas = [{ phx_ref = "uKpvL9lGfw0=", payload = { online_at = "1500725539", device = "browser" } }] },{ metas = [{ phx_ref = "RW6kKhTyRRo=", payload = { online_at = "1500725535", device = "browser" } },{ phx_ref = "RqT4Wmqtw6A=", payload = { online_at = "1500725531", device = "browser" } }] }]```
U0JBSEGHY : Maybe I am just looking at this the wrong way.
U3SJEDR96 : `yourDict |&gt; Dict.map (.metas &gt;&gt; List.length) |&gt;Dict.toList |&gt; List.indexedMap (\(name, numberofMetas) -&gt; Html.text &lt;| name ++ " has " ++ toString numberofMetas ++ " metas")`
U3SJEDR96 : `listDefault` is really just `Dict.values`, apparently
U0JBSEGHY : Oh true it is.
U27KX9MEJ : <@U3SJEDR96> yes, that's bad to define a way like this because i have a lots functions like makeWidgetLoad And every time when i want to add new widget i need to rewrite every function written this way;(I have pretty big app and wanted to make refactorring simplier. But thank you in advance i very appriciate Elm community. Elm really cool language
U27KX9MEJ : <@U3SJEDR96> That would be a great feature if we can to do pattern matching with union types and if we will get ability to use types as variables in this patterns? :slightly_smiling_face:
U0JBSEGHY : What does `&gt;&gt;` do?