

U3CUFAX4H : well, no, I was only remarking on your type sig.

U57KYFW67 : I haven't touched Haskell in a while. I should learn the new Stack package manager so I can hack on the elm compiler.

U3CUFAX4H : null decoder `find this or if err then find this ...` if you chain it together.

U3SJEDR96 : I have a feeling we're talking about completely different things, here

U4872964V : <@U3SJEDR96> maybe have `ErrorType s = ... | Custom s` too :slightly_smiling_face:

U4872964V : when we're going all custom on the errors

U3SJEDR96 : Yeah, was considering that, too. `Decoder e a` seems like it could work, actually

U4872964V : Now I'm sort of thinking about how to have extensible union types

U57KYFW67 : boo hiss at extensible union types! The goodness of having a fixed set of constructors is that our `case` statements can be complete.

U4872964V : well, maybe this is not for <#C192T0Q1E|beginners> :slightly_smiling_face:

U641LDZFU : Morning everyone :slightly_smiling_face: I'm looking for a function that groups a list of records based on a field value i.e. ``[{id= 1}, {id= 2}, {id=1}]` should output `[{id=1}, {id=1}], [{id=2}]` ``

U641LDZFU : I have had a wee look at the docs for list, and list.extra

U641LDZFU : but nothing jumped out

U1NME8MS8 : Did someone tried to implement lazy loading images before in elm? I fear I need some JS for that

U57KYFW67 : <@U641LDZFU> I think you want this:
<<http://package.elm-lang.org/packages/elm-community/list-extra/6.1.0/List-Extra#groupWhile>>

U641LDZFU : Thanks <@U57KYFW67>

U641LDZFU : List.Extra.groupWhile (\a b-> a.id == b.id) [{id= 1}, {id= 2}, {id=1}][[{ id = 1 }],[{ id = 2 }],[{ id = 1 }]] : List (List { id : number })

U641LDZFU : So I would have to sort the list based on the id first for this to be effective?

U57KYFW67 : I'm actually looking at that.... the name makes it sound like it....

U57KYFW67 : In Haskell, the function is called `groupBy` and I don't think it requires sorting

U57KYFW67 : I'm wondering if that's why the name is different

U641LDZFU : There was a nice discussion <<https://github.com/elm-community/elm-list-extra/issues/38>>

U3SJEDR96 : There is a `groupBy` that gives you a dict in `dict-extra`. Doing `values` on that should give you what you need.. `yourList |> Dict.Extra.groupBy .id |> Dict.values`

U57KYFW67 : Hmm. Interesting discussion. I'm too tired to dig into the details right now, but I can imagine Haskell doesn't do this kind of thing right :stuck_out_tongue:

U641LDZFU : I will have a go with that, thanks <@U3SJEDR96> !

U641LDZFU : ``[{id= 1}, {id=2}, {id=1}] |> Dict.Extra.groupBy .id |> Dict.values[[{ id = 1 },{ id = 1 }],[{ id = 2 }]] : List (List { id : number })
``

U57KYFW67 : I found a typo in a docstring. I corrected it, but I want to make a link. Is there a way to do that?

U641LDZFU : Magic, thanks again!

U57KYFW67 : It's a technical term, and it deserves a link to Wikipedia, I think

U3SJEDR96 : <@U57KYFW67> `[name of link](url)`

U3SJEDR96 : alternatively, `[name of link]` and at the bottom, `[name of link]: url` (i.e. a sort of reference that is inlined while rendering)

U57KYFW67 : yeah. Thanks :slightly_smiling_face:

U57KYFW67 : the name `groupByTransitively` is bad :disappointed: lol

U57KYFW67 : Yeah. I'm not a fan of how these functions look. I'm not sure it could be improved, though.

U64MK7215 : is there any sample json request code built in elm?