U1CE9DL9H : that json looks more like a record

U23SA861Y : depends if you have for knowledge of the keys

U5LFUHH19 : Right, so the return is a list of dict/records that have loads of data about the objects (biological samples) that were found. I'm struggling to understand how to deal with this, because some of the values are strings and some are numbers. The fields vary and new ones could be introduced at any time. If I define some sort of Record, would I have to account for every field/datatype?

U23SA861Y : what are you ultimately doing with this data?

U5LFUHH19 : Users will search for sample using a form. I want to display the results in a table and also use the returned data to narrow the search parameters. E.g., if "chlorophyll" ranges from 0-100 for all samples in the db but then only from 0.5-2.3 for the subset returned, I will want to show that to the users as a pair of min/max inputs. Likewise, the "Longhurst province" might just be a subset of 2-3 string values, in which case I'd construct a multi-select.

U5LFUHH19 : Looking at Json.Decode.oneOf, it looks like I'd have to coerce everything to a string?

U23SA861Y : no

U23SA861Y : do you need to recurse, as in some of your keys contain dicts or are you fine with strings and ints

U5LFUHH19 : Strings and numbers (ints and floats) are fine.

U5LFUHH19 : I should actually remove that one dict/object as it's the key from Mongo

U23SA861Y : ```type Type = StrType String | IntType Int | ValueType Dec.Value

```
decoder : Dec.Decoder (Dict.Dict String Type)
decoder =
   [Dec.string |&gt; Dec.map StrType, <http://Dec.int|Dec.int> |&gt; Dec.map IntType, Dec.value |&gt; Dec.map
ValueType]
      |&gt; Dec.oneOf
      |&gt; Dec.dict
```

something like this then

U23SA861Y : Theres probably a way to structure this without the `ValueType` so that it just omits the key if it's not an int or string

U5LFUHH19 : Thanks! I'm trying to work this into my code and understand all the implications.

U23SA861Y : If you want to store two different types in a Dict then you need an alternation that allows you check whats stored

U5LFUHH19 : <@U23SA861Y> It took me a bit to fix all my types, but, true to form, when it compiled it, Elm worked and gave me what I wanted!

U5LFUHH19 : I'm still unsure what the "ValueType" holds. Is this a catch-all for things that aren't strings or ints?

U5LFUHH19 : I'm seeing my floats are wrapped as ValueType.

U23SA861Y : yeah, its going to decode as a Json.Value

U23SA861Y : which is a generic container for json stuff

U23SA861Y : you could also probably do something this Maybe types

U5LFUHH19 : Cool, I just added FloatType and it worked.

U23SA861Y : If you don't like the values then you can do something like this```

```

type Type
   = StrType String
   | IntType Int
   | FloatType Float



decoder : Dec.Decoder (Dict.Dict String Type)
decoder =
   [ Dec.string |&gt; Dec.map (StrType &gt;&gt; Just)
   , <http://Dec.int|Dec.int> |&gt; Dec.map (IntType &gt;&gt; Just)
   , Dec.float |&gt; Dec.map (FloatType &gt;&gt; Just)
   , Dec.succeed Nothing
   ]
      |&gt; Dec.oneOf
      |&gt; Dec.keyValuePairs
      |&gt; Dec.map (List.filterMap (\( n, v ) -&gt; Maybe.map (\x -&gt; ( n, x )) v))
      |&gt; Dec.map Dict.fromList
```

```

U23SA861Y : <@U5LFUHH19> ^
U5LFUHH19 : That's going to take me a while to parse!  Thanks.
U5LFUHH19 : head == asplode :slightly_smiling_face:
U23SA861Y : the only real complicated part is what I"m doing there with filterMap
U23SA861Y : honestly I'm playing code golf here, that could easily be broken out to make it more readable
U5LFUHH19 : I have to head home, but thanks, again, for your help!
U23SA861Y : no problem <https://ellie-app.com/3xk5T6PDDNLa1/0>
U2LAL86AY : <@U2Q3QE54G> have you tested the same functionlity in js -  to make sure that what you have there is not a browser problem ?!
U0JL9RPC4 : Hi! Do you think it would possible somehow to use Elm as a desktop native application in order to get access to local files, directories, etc?
```