U0U6ML22H : apparently, I cannot create a Dict from a `List (Date.Date, Float)`, which, why not, but the message is not at all that... is it?
U4872964V : well, it depends on how you look at it I suppose, there is certainly room to clear away cruft and say that the mismatch is that `Date.Date` is not `comparable`
U3SJEDR96 : It's not _wrong_. It's not super clear, but it's not exactly wrong :smile:
U3SJEDR96 : but yeah, `Date` isn't `comparable`
U0U6ML22H : I figured that was the problem, but certainly not thanks to the message ^^
U0U6ML22H : (btw, Date could totally be comparable ^^)
U4872964V : The Elm compiler generally has such good error messages so you get disappointed at messages like this, but coming from Haskell this message is super clear :slightly_smiling_face:
U57KYFW67 : "The following difference equation has no solution in the category of small Haskell types: ..." <- the typical Haskell error message.
U0U6ML22H : ^^
U3SJEDR96 : augustin82: if it were special cased - sure. But special casing = more magic, and that's not always good, either :wink:
U37HUSJ4R : How would you write a json decoder for a field with dynamic keys, but you care about the key? For example I have a type```
type alias OpeningHours =
    { verb : String, message : String }
```

and the json looks something like `{"play": "foobar"}` or it could be `{"say": "baz"}`

U37HUSJ4R : where `verb` would be either `play` or `say`
U1CE9DL9H : decode it as a dictionary
U3SJEDR96 : that, or `keyValuePairs string |&amp;gt; List.map (uncurry OpeningHours)`
U1CE9DL9H : yes that is better. the dict approach will give problems when using the same key for multiple values
U37HUSJ4R : interesting, do you have an example of using `keyValuePairs` with a decoder <@U3SJEDR96>
U1CE9DL9H : there is a `Decode.map` missing I believe
U3SJEDR96 : urrgh, yeah, there is :smile:
U0U6ML22H : <@U37HUSJ4R> funny, I'm writing that kind of decoder as we speak :smiley:
U0U6ML22H : ```decodeStatuses : D.Decoder (Dict.Dict Float SM.SpotStatus)
decodeStatuses =
    D.keyValuePairs (D.string |&amp;gt; D.andThen SC.statusDecoder)
        |&amp;gt; D.map (List.map (Tuple.mapFirst stringToFloat))
        |&amp;gt; D.map (Dict.fromList)
```

U3SJEDR96 : <https://ellie-app.com/3MzHwFpBZpLa1/0> for a quick example
U0U6ML22H : of course, <@U3SJEDR96> is going to come and have a better solution ^^ thanks for `uncurry`, good to know!
U37HUSJ4R : great
U37HUSJ4R : thanks <@U3SJEDR96>
U0U6ML22H : you're welcome ^^
U3SJEDR96 : <@U0U6ML22H> that `stringToFloat`, tho... >.>
U0U6ML22H : what's wrong with it?
U3SJEDR96 : it's throwing away errors, as opposed to sometihng like this:
<http://package.elm-lang.org/packages/elm-community/json-extra/2.3.0/Json-Decode-Extra#dict2>
U0U6ML22H : ah, that! yeah, you're right :smiley: I've been iterating several times on this code, weighing implementations, going back and forth with Dict or List, so it's a quick'n'dirty version for now