

U0EUHKVGB : Yes. You cannot have varying keys in a record in Elm, because it makes the types impossible to figure out.

U0EUHKVGB : You can have multiple different records

U0EUHKVGB : But that's different from varying keys.

U5TQ5FZA5 : <@U0CLDU8UB> yeah - the keys are String and the values are String | Dict

U0CLDU8UB : I think you might be confusing how Dicts and records work. Records have certain named fields, with specific types per field. Dicts are collections of for key and value pairs, where the keys need to be of one single type (strings are fine), and the values need to be of a single type, but that includes the union type <@U4872964V> showed earlier.

U4872964V : ok, so then``

```
type alias YourDictType = Dict String EitherStringOrDict
``
```

U0CLDU8UB : yep, that

U5TQ5FZA5 : <@U0CLDU8UB> yep - I think you're right there!

U5TQ5FZA5 : Thanks, <@U4872964V> - I'll try that :slightly\_smiling\_face:

U3YA74FQD : What's the best strategy when elm is reusing an input node (and erroneously displaying the old value)?\* Should I try to force a new empty value to the input?

\* Should I look into `Html.Keyed`? I feel that this would be incompatible with packages like `rundis/elm-bootstrap`

U4RR7KX45 : Hello elmers :smile: How can I decode a json list? I have the following function:

``

```
fetchRandomQuotes : Http.Request (List String)
```

```
fetchRandomQuote =
```

```
    Http.get randomQuoteUrl decodeQuoteUrl
```

```
decodeQuoteUrl : Decoder (List String)
```

```
decodeQuoteUrl list =
```

```
    <http://Decode.at|Decode.at> [ "" ] list
```

``

U4RR7KX45 : I couldn't find a good tutorial that explains this so I sort of followed the cats git example

U4RR7KX45 : but couldn't get far

U48AEBJQ3 : <@U4RR7KX45> What is in the list?

U4RR7KX45 : at the moment I'm just hard coding it on the api

U4RR7KX45 : and it's just an array of strings

U48AEBJQ3 : `Decode.list : Decoder a -> Decoder (List a)` So you need to hand it a decoder for each item in the list and it will apply it across the whole list.

U4RR7KX45 : but later on, I want to use the api that returns objects:``

```
{"ID":1368,"title":"Thomas Manss","content":"&lt;p&gt;Decoration is just make-up for the wrinkles of the idea.
&lt;p&gt;\n","link":"https://quotesondesign.com/thomas-manss-2V"}
```

``

U48AEBJQ3 : Then it is `Decode.field "myField" (list string)`

U48AEBJQ3 : Later on you will write a decoder for a single object and go `Decode.field "myField" (list myObjectDecoder)`

U4RR7KX45 : what does myField refer to in my case?

U4RR7KX45 : a specific field like title?

U48AEBJQ3 : A field in the JSON object. If your top-level is a list of strings, you \*only\* use `(list string)`

U48AEBJQ3 : Right, so in the object you gave as an example, you might have `Decode.field "ID" int`

U48AEBJQ3 : If you want to decode the entire thing:

``

```
Decode.map4 mkThing
```

```
    (Decode.field "ID" int)
```

```
    (Decode.field "title" string)
```

```
    (Decode.field "content" string)
```

```
    (Decode.field "link" string)
```

``

U4RR7KX45 : I think I have a mistake in the import too, as it says Cannot find variable Decode.field

U48AEBJQ3 : You can probably just use `field`