

U6D1E91QV : there's no json involved there, i am merely trying to copy the api  
 U0LPMPL2U : Ah, so you have many `Result` values that you want to combine together?  
 U6D1E91QV : in a way -- and collect in a record  
 U6D1E91QV : can i paste stuff in here?  
 U0LPMPL2U : sure that would be helpful :slightly\_smiling\_face:  
 U6D1E91QV : ```type FieldValue raw val err  
     = Empty  
     | Partial raw  
     | Invalid raw err  
     | Valid val  
 ...

U6D1E91QV : that's a field type for a for field, which gets updated upon input and may or may not be valid  
 U0LPMPL2U : OK  
 U6D1E91QV : ```extractField : FieldValue raw val err -> Result (List err2) val  
 extractField =  
     value &&> Result.fromMaybe []  
 ...

U6D1E91QV : this will turn any FieldValue into either a `val` or a list of errors  
 U6D1E91QV : example use```  
 extractUsername form username = (extractField &&> .username)  
 ...

U6D1E91QV : given some sort of form record with a field `username`, this will extract said value as a result  
 U0LPMPL2U : Where does the list of errors come from? Doesn't the `Invalid` tag only allow one error?  
 U6D1E91QV : (excuse the typo please)  
 U6D1E91QV : the list is always empty initially  
 U6D1E91QV : the errors collected are not on a field, but on a form level  
 U6D1E91QV : i mean to add additional validators that validate on a form level, e.g. can check if password1 and password2 are the same  
 U0LPMPL2U : hmmm... since `extractField` only deals with a single field, shouldn't it only return a single error?  
 U6D1E91QV : but you're right, that needs to go  
 U6D1E91QV : i think even a Maybe would be enough here. anyway, the idea is to have a way of transforming a Form into a model  
 U0LPMPL2U : makes sense  
 U6D1E91QV : the matching model would be:```  
 type alias Model = { username: String, password: String }  
 ...

U6D1E91QV : at that point, each field is a valid instance of a possibly valid value  
 U0LPMPL2U : So you'd like to do something like `Field.map2 Model field1 field2` ?  
 U6D1E91QV : yes, sort of, i believe  
 U6D1E91QV : field1 and field2 would probably be extraction functions?  
 U6D1E91QV : or Decoders, so to speak  
 U0LPMPL2U : so extracting each field already gives you a `Result` right?  
 U6D1E91QV : at the moment, yes. you got me thinking that a `Maybe` is enough.  
 U0LPMPL2U : Both `Result` and `Maybe` define a `map2` function  
 U6D1E91QV : basically, the decoding needs to only work if every field is `Valid ...`