

U3SJEDR96 : When dealing with the url-parser library, I think it's best not to worry about the type signatures too much :x

U0D4VR7BP : the structure is `Parser x y` if that helps you making sense of the arrows

U2SR9DL7Q : oh. <@U0D4VR7BP> i see it now

U0D4VR7BP : in this case `Parser fnTakingRoute result`, it seems like

U0D4VR7BP : I don't know what it does, I'm just guessing at the meaning

U2SR9DL7Q : to the documentation I go then. I've got it all working for my own site after following the example, but this is the one part where I feel myself not understanding exactly what's going on.

U2M4VPZ9D : I pushed the code at <https://github.com/kafecho/elm-beats>

U3X6H8X8W : Hi everyone

U3X6H8X8W : I have a small questions about how you define your records types for entities that you can create, list, edit remotely.

U3X6H8X8W : I have a `Product` type with several fields, but this record's `id` field is only present when this is a persisted entity.

U3X6H8X8W : ```type alias Product = { id : Int
, name : String
, description : String
, vendorId : Int
, unitPriceCents : Int
, internalUnitPriceCents : Int
... }

U3X6H8X8W : do you model the `id` as a `Maybe` and use the same type for persisted and new products?

U3X6H8X8W : or should I create a separate type for this?

U3X6H8X8W : or maybe like a union `type Product = PersistedProduct IdType CommonData | NewProduct CommonData`?

U3X6H8X8W : I want to use the type system as much as possible and don't know if I'm overdoing it :see_no_evil:

U2SR9DL7Q : It feels a little dubious IMO to have a `Product` type like this with a maybe id. I feel like its trying to make a product and a _template of a product_ be the same. But I may be misunderstanding... In what case would it not be a persistent entity?

U3X6H8X8W : a non persistent entity is for instance a `Product` being created from a form that I have to convert to a JSON object and then post it to the API to see if it can be saved.

U3X6H8X8W : for instance I have a button that says `Create new product` and shows me inputs to fill all the data.

U3X6H8X8W : For that I have like a separate form type? I don't know how to find examples of how to do it.

U3X6H8X8W : Another case is that I have a real `Product` that I decoded from an API call and now I want to update it, this new data that I haven't sent to the server is not really a `Product` because I don't even now if it's all valid according to the API

U2SR9DL7Q : Ahhh. I see. I wouldn't call that a product then. Perhaps a product request or something similar. My worry with the maybe Int is that you would need to go to every place where the id is used and ensure you handle the maybe case. But after this initial form request and sending the json, there's no reason why you would want a product without an id.

U2SR9DL7Q : So you would have product request of product update, and such, but they would be kept separate from an actual product, which is a persistent, validated thing

U0FP80EKB : I've done where you have the union type and it works okay `type Product = Persisted Id Data | NotPersisted Data`

U0FP80EKB : It also helps in that you don't run into situations where you have an id, but think you don't

U0FP80EKB : Make it very clear

U0FP80EKB : Also, can do something like `type alias ProductData = {...}`
and
`type Product = Product Id ProductData`
Then you can just build up the product data in your form handler, post it, then take the return value and convert it to `Product`

U3X6H8X8W : I saw something like this in elm-spa-example: ```type alias CreateConfig record = { record | otherField : }`. All ids in my API are `Int` so maybe having something like `type alias Persisted x = { x | id = Int}`?

U3X6H8X8W : <@U0FP80EKB> something liek what you mention is what I was thinking of

U0FP80EKB : I don't like extensible records like that, personally. They always end up causing me headaches, because they are a form of inheritance with all the problems that come with that.

U0FP80EKB : I'd say pick one, use it, then identify the pain points and migrate to another solution to fix those pain

points. It is easier in Elm to refactor than in other languages

U2SR9DL7Q : <@U0FP80EKB> that last example you wrote looks pretty good, and it's easy to see what the data actually is.

U0FP80EKB : Every time I use an extensible record, for example, and I do try using them (currently experimenting with using one in our current system), I end up finding they aren't worth the headache, so I migrate away from them

U3X6H8X8W : Exactly <@U2SR9DL7Q> Also I prefer to use the most "voted" option considering other people experience and probably I never have to pick the other one

U0FP80EKB : Also, you might want to look at using ``type Id = Id Int`` rather than ``type alias Id = Int``

U0FP80EKB : We started doing this a long time ago and it has been very nice

U638BAUJZ : Can someone please help me resolve a compiler

error?<<https://stackoverflow.com/questions/44989599/how-do-i-initialize-a-value-thats-based-on-a-recursive-type-alias>>