U23SA861Y : <@U40QW928G>, do you got a repo to point at so we can get a better feel?
U40QW928G : I can create one
U0LPMPL2U : Imagine two files```
module User.Commands exposing (fetchUsers)

fetchUsers : (a -&gt; msg) -&gt; Cmd msg
fetchUsers tagger =
  -- fetch users and tag with tagger
```

```
module Main exposing (main)
import User.Commands exposing (fetchUsers)

type Msg = FetchUser | UserFetched User

update : Msg -&gt; Model -&gt; (Model, Cmd Msg)
updage msg model =
  case msg of
      FetchUser -&gt;  (model, fetchUsers UserFetched)
      -- rest
```

U0LPMPL2U : <@U5LFUHH19> are you using `elm-community/json-extra` ?
U5LFUHH19 : Ah, no.
U40QW928G : ok I have that
U40QW928G : ```fetchUser : Cmd msgfetchUser =
    Http.send FetchUsers userEndpoint```

U0LPMPL2U : right. Instead of hardcoding `FetchUsers`, pass it in as an argument
U0LPMPL2U : that way `fetchUser` function has no dependency on the `Msg` type
U40QW928G : oh ok
U0LPMPL2U : ```fetchUser : (a -&gt; msg) -&gt; Cmd msg
fetchUser tagger =
    Http.send tagger userEndpoint
```

U40QW928G : ok but now
U5ABF3BH7 : In my company, we are transitioning from Rails to Elm for the FrontEnd. So far, things have been built thinking only about the front end, building models that make sense for the front end, and without thinking too much about the back end. Now I have the task of refactoring so that we have a better separation between data and views. I am having a hard time because it is all tightly connected in several layers. For example, one view touches different database models, and the same database model is represented in different views. I am a bit confused as what is the best way to encapsulate that in the view. Let's say I have an Address model, an Organization model and a person model (I have many more but it is to simplify). Both the organization and person have a contact information which in turn has phone numbers (which are all models). And all those fields are in one view. A person and an organization have an address.How would you design your view model to reflect that complexity, making it easy to encode and decode. I know I am to separate your data from the view, but I am hesitant on what is the right way to go about it. Any advice? Sorry for the long paragraph.
U40QW928G : It's yelling at me for this function type
U40QW928G : ```userEndpoint : Http.Request (List User)
userEndpoint =
    Http.get (baseUrl ++ "/users") decodeUsers
```

U40QW928G : ```Function `send` is expecting the 2nd argument to be:

    Http.Request a

But it is:
```

    Http.Request (List User)
```

U0LPMPL2U : Can you change the signature to `fetchUser : ((List User) -&gt; msg) -&gt; Cmd msg` ?
U23SA861Y : Think it needs to be a `(Result Error (List User)) -&gt; Cmd msg`
U40QW928G : ```The 1st argument to function `send` is causing a mismatch.

20|     Http.send tag userEndpoint
                  ^^^
Function `send` is expecting the 1st argument to be:

    Result Error a -&gt; msg

But it is:

    List User -&gt; msg
```

U0LPMPL2U : ah yes
U23SA861Y : sry `(Result Error (List User) -&gt; msg) -&gt; Cmd msg`
U23SA861Y : look at the type signature for send and replace `a` with `List User`
U40QW928G : yeap that worked
U40QW928G : now idk what to do with the update