

U23SA861Y : if they are so tightly integrated the question becomes should they  
 U0LPMPL2U : would it make sense to pass in the message for your commands?  
 U0LPMPL2U : that's how the core library deals with commands, they let you pass in your own messages  
 U23SA861Y : Yeah a context or function parameter to configure the call  
 U31A7HG1E : I just ran into this very same problem and ended up doing exactly as <@U0LPMPL2U> suggested  
 U40QW928G : I'm not sure I'm seeing your solution  
 U0LPMPL2U : ``myUserCommand : (a -&gt; msg) -&gt; Cmd msg  
 myUserCommand tagger =

...

U5ABF3BH7 : Would any more advanced Elm people be interested in giving 1 or 2 hours of tutoring to give some guidance on how to refactor a project? I live 1 hour South of San Francisco.  
 U0LPMPL2U : `myUserCommand` no longer has any dependency on a particular `Msg` type, you pass it in instead  
 U0LPMPL2U : this is the approach used by the core library, e.g `Random.generate`  
 U0LPMPL2U : ``generate : (a -&gt; msg) -&gt; Generator a -&gt; Platform.Cmd.Cmd msg

...

U23SA861Y : it's actually used in almost every core lib function  
 U0LPMPL2U : right because the core library \_can't\_ depend on your custom `Msg` type  
 U0LPMPL2U : <@U5ABF3BH7> have you been to the SF Elm meetup? I hear it's really good :slightly\_smiling\_face:  
 U5ABF3BH7 : I have, it was good, I should have asked then.  
 U5ABF3BH7 : I went to the last 2  
 U23SA861Y : You could hop on flight up to vancouver, Canada. :stuck\_out\_tongue:  
 U40QW928G : oh ok interesting  
 U5ABF3BH7 : <@U23SA861Y> :slightly\_smiling\_face:  
 U5LFUHH19 : As I transition to Elm from Perl, I'm finding my JSON encoding is wacked. From my existing Perl JSON API, what should be Ints have `""`s around them, so they look like strings. Is there a way to decode them with a `String.toInt` kind of thing so I can match the JSON to my (proper) objects?  
 U0LPMPL2U : Yes but you'll have to worry about `Result` because the string might not be a valid integer  
 U0LPMPL2U : You can say:

...

```
stringInt : Decoder (Result String Int)
stringInt =
  JD.map String.toInt JD.string
```

...

U0LPMPL2U : You could then use `andThen` to check for `Ok` or `Err` and convert that to `JD.succeed` or `JD.fail`  
 U0LPMPL2U : or you can use `Json.Decode.Extra.fromResult` which does that for you  
 U0LPMPL2U : e.g.``  

```
stringInt : Decoder Int
stringInt =
  string |&gt; JD.andThen (String.toInt &gt;&gt; JDE.fromResult)
```

...

U40QW928G : I guess I'm just not understanding where to put messages  
 U40QW928G : or for that matter how to import them  
 U40QW928G : I try `import User.Commands exposing (UserMsg)`  
 U23SA861Y : if you did that your messages would be imported by update but not user  
 U40QW928G : but when I try to put `FetchUser` in the case it doesn't work  
 U40QW928G : ``Cannot find pattern `FetchUsers`

```
26|     FetchUsers -&gt;
    ~~~~~
```

Maybe you want one of the following?

```
User.Commands.FetchUsers
```

...