U2PTZFKFX : thanks <@U32BTDW4U> what about http requests and manipulating the json data?
U2PTZFKFX : ok
U0Z9TPK7S : <@U2PTZFKFX> probably the easiest way to manipulate JSON data is to decode it into Elm records and lists, manipulate those as you'd do normally in the language, and re-encode them to JSON.
U2PTZFKFX : ok
U0Z9TPK7S : Does that answer your questions?
U2PTZFKFX : somehow, let me read what is posted first,
U0Z9TPK7S : <@U23SA861Y> I stand corrected. I see now that your suggestion of making a more generic view does make sense. Again thank you for your input.
U0J8D9M2P : Exactly
U0J8D9M2P : <@U635238TG> Thank you
U5ZC6V535 : Hi i am trying to separate my top level messages to sub messages and hence i did:
```
type GeneratorMsg
    = BoidsGenerated (List Boid)
    | ColoursGenerated (List Color)


type Msg
    = Tick Time.Time
    | UpdateWorld Window.Size
    | GeneratorMsg
```

However, in my main update function when I use the *BoidsGenerated* message Elm thinks that it is of type *GeneratorMsg* which is correct. In the same time though - in my mind -, it is of type *Msg*.

Is there a way to be able to handle *Msg* and *GeneratorMsg* interchangeably? Basically, i want to split my update function to smaller functions but I want everything that has to do with generated stuff to be handled by 1 separate function. Then that function will have cases for *BoidsGenerated* and *ColoursGenerated* msgs. --- thanks

U0Z9TPK7S : <@U5ZC6V535> You should declare those as```
type Msg
    = Tick Time.Time
    | UpdateWorld Window.Size
    | GeneratorMsg GeneratorMsg
```

U0Z9TPK7S : But this is going to be confusing; I would recommend to change `type GeneratorMsg` into `type GeneratorOutcome`.Then your update function can be
```
type GeneratorOutcome
 = BoidsGenerated (List Boid)
 | ColoursGenerated (List Color)

type Msg
 = GeneratorMsg GeneratorOutcome
  | ...

update msg model =
  case msg of
    GeneratorMsg generatorOutcome ->
      generatorUpdate generatorOutcome model

    ...
```

U0Z9TPK7S : Does it make sense?
U5ZC6V535 : <@U0Z9TPK7S> Yeah, that actually makes sense :slightly_smiling_face: Thanks.
U0Z9TPK7S : :smiley:
```

U2LAL86AY : <@U5ZC6V535> i use a `word pattern` for delegating messages to sub modules:```
type Msg =
  | MsgFor_Generator GeneratorMsg
  | MsgFor_SomethingElse SomethingElseMsg
  | Click

type GeneratorMsg = -- no need for GeneratorOutcome, or other variations - because sometimes "outcome" si a word that doesn't work semantically, and you waste time thinking : "what would be a better naming ?!"
    BoidsGenerated (List Boid)
    | ColoursGenerated (List Color)
```
It's how i keep my mind and msgs consistent. :smile:

U2LAL86AY : and in your elm debugger will look like:`MsgFor_Generator ( GenerateSomething { seed: 1112112 })`

U24HQ3RJ7 : hi.. i am new to CSS and have to make a web / mobile app .. I am using Elm for it. :slightly_smiling_face: Has the community converged on some particular css library ? elm-css vs elm-style-elements vs something_else ..

U0Z9TPK7S : <@U24HQ3RJ7> Hi! Welcome!
U0Z9TPK7S : elm-style-element is still in its infancy, the only complete and solid CSS option right now is elm-css
U24HQ3RJ7 : <@U0Z9TPK7S> thanks.. since this is side project, i will give style-elments a try.. :slightly_smiling_face:
```