U2D5SAEMN : I could place the numbers in a separate property on the Event, e.g. `payload: Maybe Int`? Then the EventType union type doesn't need the extra parameter.
U0CLDU8UB : I have an idea that would work rather nicely, but involves understanding extensible records. Your idea works too, and it's not bad at all.
U0CLDU8UB : I'll update the Ellie to the idea that I have just as a reference, in case you're interested
U2D5SAEMN : really? wow, thanks! :raised_hands:
U4872964V : A simple option is to have a separate union type for filters, like `BirthdayFilter` etc, and then have a filter function that does `case` on that type. More boilerplate, but type safe
U2D5SAEMN : How would the filter function work then?
U0CLDU8UB : Yeah, the idea that I had is actually much more cumbersome in code than I thought it would be, so let's scratch that. :slightly_smiling_face:
U4872964V : So, I'd do like this (but this is just a preference)
```
filterEvent filter event =
  case (filter, event) of
    (ActiveFilter, Active) -&gt; True
    (ActiveFilter, _) -&gt; False
    (BirthdayFilter, Birthday _) -&gt; True
    (BirthdayFilter, _) -&gt; False
    ...
```
I generally do it like this so I don't have to have a "catch all"

U4872964V : or the arguments in the other order, maybe, for your usage of the filter
U2D5SAEMN : `event` would be the `eventType` property of the event record, I guess?
U4872964V : another option is to do it like this
```
filterEvent filterList event =
  case event of
    Active -&gt; List.member ActiveFilter filterList
    Birthday _ -&gt; List.member BirthdayFilter filterList
    ...
```
sorry if I've not understood your problem correctly

U2D5SAEMN : ohanhi: Thanks anyway!
U2D5SAEMN : nice, that would support multiple filters as well. Thanks a lot!
U4872964V : Generally I find that making specific types pay off in simplicity and type safety, even if it leads to more code
U4872964V : you're welcome!
U4872964V : you could of course break it out to make a function `: Event -&gt; EventFilter` and make that `filterEvent` function more general, it's just refactoring at that point
U2D5SAEMN : Sorry, I don't understand what you mean by that? :slightly_smiling_face:
U4872964V : so, you'd make something like
```
eventFilterFor : Event -&gt; EventFilter
eventFilterFor event = case event of
  Active -&gt; ActiveFilter
  Birthday _ -&gt; BirthdayFilter
  ...
```
it depends on how complicated your filters should be, the previous solution allows for different kind of filtering based on the event. So it's refactoring/preference/etc

U2D5SAEMN : ah, I see. That would be useful when filtering is done in multiple places, I think.
U5GMHEAHY : Hi there! So I've been working on a project that, at some point, navigates to a page that communicates with the server using a websocket. This is to get live updates of the results from the server (it's performing a task). I'm experiencing a problem with this:
- When using Elm Navigation, if the route and its query parameters are the same, it does not trigger the update msg that I implemented (which sends a GET request to the server asking for previous results, which works fine). It does, however, trigger upon switching routes (editing the query parameter back and forth or simply navigating to a different page) so I assume that Navigation does not attempt to perform the specified actions if the conditions are the same, even if it's a refresh request.

U5GMHEAHY : What I would like to happen is that the update msg triggers upon refresh. Any suggestions?
U57KYFW67 : I forget the hoogle-search for Elm. What was the URL again?
U4872964V : It's linked at the front page of elm packages: <http://klaftertief.github.io/elm-search/>
U57KYFW67 : ty
U57KYFW67 : Is there a way to get the screen (or top-level element) height and width?
U2JA29EKF : <@U57KYFW67> : <http://package.elm-lang.org/packages/elm-lang/window/1.0.1/Window>
U57KYFW67 : thanks