

U0EUHKVGB : When deploying?

U52GHJJTU : I've moved `node_modules` elsewhere using the `NODE_PATH` env variable

U0EUHKVGB : You should compile your Elm files before deploying

U52GHJJTU : docker-based development environment

U52GHJJTU : Our webpack runs in an environment where all source files and their directories are read-only

U0EUHKVGB : Simple fix: put `elm-package.json` into your writable directory

U0EUHKVGB : then run webpack from there.

U0EUHKVGB : `source-directories : "../../some_read_only_dir"

U52GHJJTU : Ah, that would do the trick. Running `elm-package` will be somewhat harder, but with this setup running `npm` is also hard :slightly_smiling_face:

U52GHJJTU : Thanks :thumbsup:

U5FU80S06 : I want to create a module for serving unique IDs, but cannot figure out how. I know this is static:
 ```module ID exposing (ID(..), new)  
 type ID = ID Int

count = 0

```
new =
 let
 newCount = count + 1
 in
 ID newCount``` but this is definitely wrong: ```new =
 let
 count = count + 1
 in
 ID count```
```

U0EUHKVGB : You cannot do that.

U0EUHKVGB : You have to pass in the previous ID

U0EUHKVGB : Alternatively, look at this package which provides a way of generating UUIDs:  
 <<http://package.elm-lang.org/packages/danyx23/elm-uuid/latest>>

U0EUHKVGB : And here for a minimal example of usage:  
 <<https://github.com/danyx23/elm-uuid/blob/2.1.0/examples/Minimal.elm>>

U5H8JJP24 : Is there somewhere an example for decoding a simple Union Type into a String ?```

```
type Theme =
 Black
 | White
 ...
```

U0EUHKVGB : <@U5H8JJP24>, you can use `toString` for that

U0EUHKVGB : Or implement a function which is `Theme -> String` if you don't want the default toString

U5H8JJP24 : <@U0EUHKVGB> Thx I will have a try

U5FU80S06 : <@U0EUHKVGB> Doesn't passing in the previous ID break information hiding? It is interesting that something trivial in JS is not so in Elm.

U3SJEDR96 : UUID's tend to involve randomness, which means "do something else every single time". It makes sense for that to not be trivial in a language where all functions should return the same result every times they're called, though it definitely makes some things harder..

U0K92QFST : <@U5FU80S06> it looks like you want the "ID" module to keep track of an internal state, which is impossible given Elm's design

U0EUHKVGB : <@U5FU80S06> Lots of things are trivial in a language with unmanaged side-effects.

U0EUHKVGB : Elm is a different language. Some things are trivial in Elm that aren't in JS

U0EUHKVGB : For example, writing code that does not break :wink: