

U153UK3FA : Oh, that's actually just a function

U153UK3FA : Declaring a type alias for a record type also defines a function to construct values of that type with the same name as the type alias

U4K9J3NUC : And is the order of arguments same as in which those were declared? And is this documented somewhere?

U3SJEDR96 : It's the same order, indeed. I don't know if that is actually clearly stated somewhere, to be honest

U3SJEDR96 : <[https://guide.elm-lang.org/types/type\\_aliases.html](https://guide.elm-lang.org/types/type_aliases.html)> there's an example there, but it doesn't specify that this is positional

U3SJEDR96 : oh, wait, it does

U3SJEDR96 : "The arguments are in the order they appear in the type alias declaration"

U4K9J3NUC : OK, cool, it was there in the docs just waiting for me to reach that page :slightly\_smiling\_face: Thanx everyone

U3SJEDR96 : happy reading! :wink:

U3SJEDR96 : interestingly, that syntax has been around since 0.9

(<<http://elm-lang.org/blog/announce/0.9#record-constructors>>)

U31FGNWCT : Hi everyone. What's wrong with the following webpack.config.js? Webpack says it can't find appropriate loader for Main.elm. I'm using webpack with gulp, btw. I did install elm-webpack-loader:``

```
gulp.task('elm', function() {  
  return gulp.src(paths.project.elmWebpack)  
    .pipe(plumber())  
    .pipe(webpack({  
      entry: '../js/elm/entry.js',  
      output: {  
        filename: 'bundle.js'  
      },  
      module: {  
        rules: [{  
          test: /\.elm$/,  
          exclude: [/elm-stuff/, /node_modules/],  
          loader: {  
            loader: 'elm-webpack-loader',  
            options: {debug: true, warn: true}  
          }  
        }  
      ]  
    })  
    .pipe(gulp.dest(paths.toolkit.jsBase));  
});  
``
```

~So far embedding elm-make in the existing build process is painful, unproductive and very expensive. Just wanted to share my overall frustration with open source.~

U31FGNWCT : Forgot to mention: paths.project.elmWebpack == ../js/elm/entry.js

U0JBSEGHY : How do I decode a single string in Elm. All the examples use a full model as the structure for decoding. Completely stumped. :confused:

U0CLDU8UB : What does your HTTP response look like?

U0JBSEGHY : {"token":"XX\_SECRET\_TOKEN\_XX"}

U180KMGRE : You just want the token then?

U0JBSEGHY : Yea I am literally just exposing some Json with the token. I'm still super new so trying to build up slowly

U0JBSEGHY : I'm running a Phoenix server and trying to connect to an API point to get the token. Then will try and figure out the rest as I go along :slightly\_smiling\_face:

U0JBSEGHY : I feel like I have gotten really close. I'll show the error that was tripping me up.

U3SJEDR96 : `field "token" string` is all you need

U3SJEDR96 : which decodes that JSON into simply the string `XX\_SECRET\_TOKEN\_XX`

U0JBSEGHY : ``fetchToken = Http.send ReceiveToken (Http.get apiEndpoint tokenDecoder)

tokenDecoder =

```
Decode.decodeString (Decode.field "token" Decode.string)``
```

U0JBSEGHY : Says function `get` is expecting `Decode.Decode a` but is getting `String -> Result String String`  
U180KMGRE : You don't want the decodeString there  
U180KMGRE : decodeString is partially applied so it then wants a string to decode, but Http.get will run the decoder when it gets the response and so on, so it only needs the decoder to run  
U0JBSEGHY : sorry I am not sure I follow  
U180KMGRE : decodeString is trying to apply the decoder to a string - you don't need that to happen - you just want the decoder `Decode.field "token" Decode.string`  
U0JBSEGHY : Was following here and various other examples. Which had variants of the solution.  
<<http://package.elm-lang.org/packages/elm-lang/http/latest>>  
U0JBSEGHY : ``The 2nd argument to function `send` is causing a mismatch.  
86| Http.send ReceiveToken (Http.get apiEndpoint tokenDecoder)  
~~~~~

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

Http.Request Model

But it is:

Http.Request String``

U153UK3FA : <@U0JBSEGHY> ReceiveToken takes a Model not a String  
U0JBSEGHY : Yea this is where I keep tripping up in all the various ways I have tried