



Building your Elm code

By: Tolga Paksoy

<https://paramander.com>

<https://github.com/tolgap>

Save → compile → refresh → **recreate state**

Current state

Write code

Reload browser

Lose application state

Recreate the state you were in to view your changes

What do we want?

Change your code

And see the changes in your browser without losing state

ClojureScript figwheel

ROI for writing reloadable code is very high

Current options for building

1. elm-make
2. elm-reactor
3. elm-webpack-loader
4. elm-live

p.s: there are many webpack alternatives that can do this as well

elm-make

- + Able to put into Makefile
 - Single compile
 - Manual
 - Zero instantaneous feedback in browser
-

elm-reactor

- + Compiles your code on save
 - + Spits compile errors into your browser
 - Becomes a hassle to use in bigger applications
 - `Html.programWithFlags`
 - Still have to manually reload the browser
 - Loss of application state
-

elm-live

- + Compiles code on save
 - + Reports errors nicely
 - Reloads browser but ports lose state
 - Doesn't cooperate as well with elm-css/external processes
 - Extra complexity with HTML files/ports/Flags (I couldn't figure it out)
-

elm-webpack-loader

- + Pluggable nature of Webpack
 - + Compiles code on save
 - + Supports reloading site without refresh
 - Requires npm/yarn to set up
 - Extra complexity of learning Webpack
 - Still loss of application state on compile errors
 - Compile errors are in terminal and JS console
-

Why is hot reloading difficult?

Requires a certain way of building code

Dependencies in separate file(s), project code in separate file

something like `bundle.js` and `app.js`

Only reload `app.js`, so deps and Elm keep running, user changes become visible.

One of Evan's focus for Elm 0.19

Subscriptions/Flags/Ports

demo

<http://localhost:8000>

A good start config for elm-webpack:

<https://github.com/benansell/elm-webpack-seed>

Until elm platform support...

External tools don't work well in monolith repo's

Serverside HTML serving doesn't work well either



Thank you

Tolga Paksoy
<https://paramander.com>
<https://github.com/tolgap>
