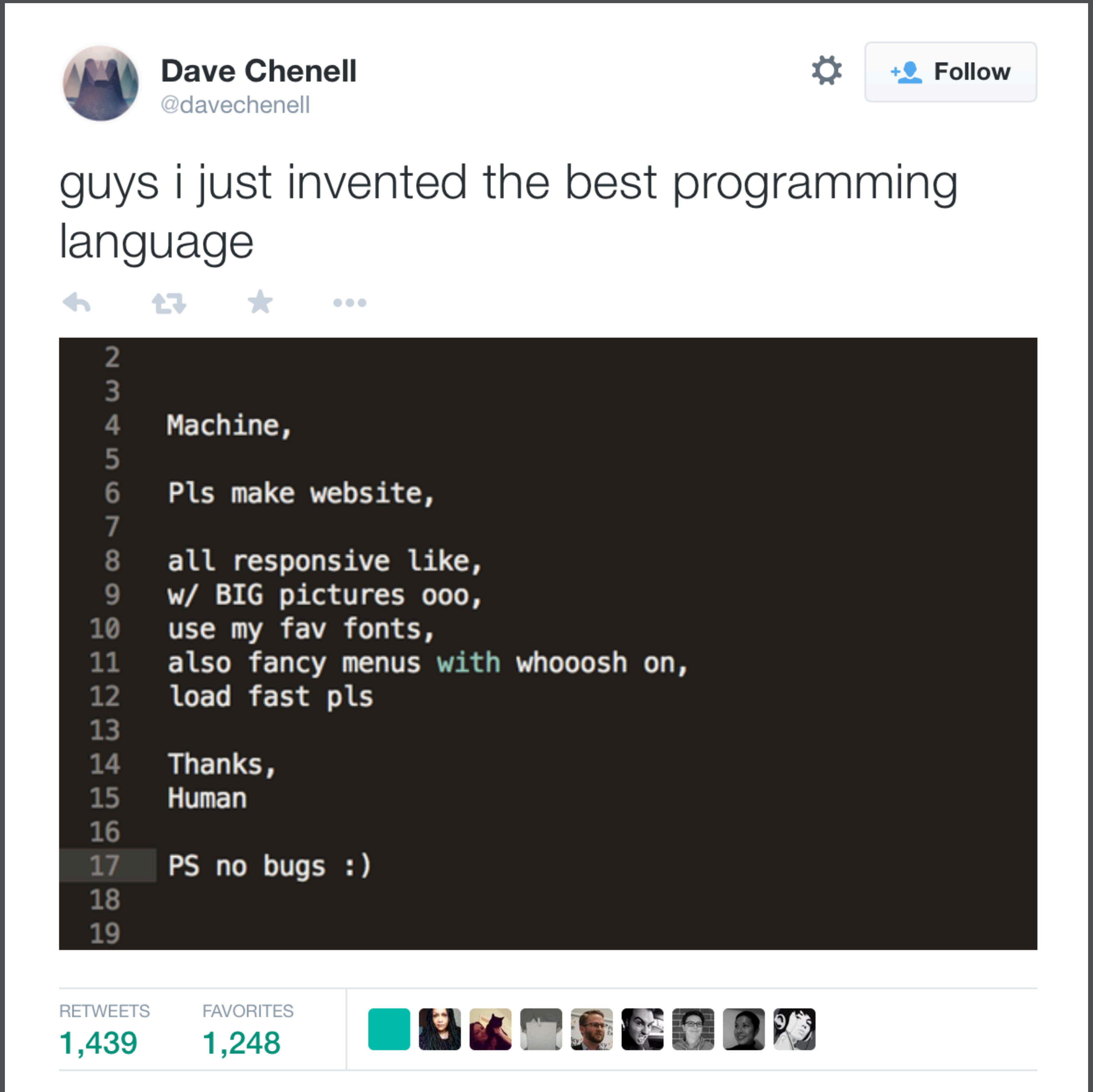
cljs + gui

simplicity yields possibilities

slides and code online (at github)

the things I dare to build thanks to clojurescript!

guis are nice, programming them not so



Dave Chenell
@davechenell

guys i just invented the best programming language

```
Machine,

Pls make website,

all responsive like,
w/ BIG pictures ooo,
use my fav fonts,
also fancy menus with whooosh on,
load fast pls

Thanks,
Human

PS no bugs :)
```

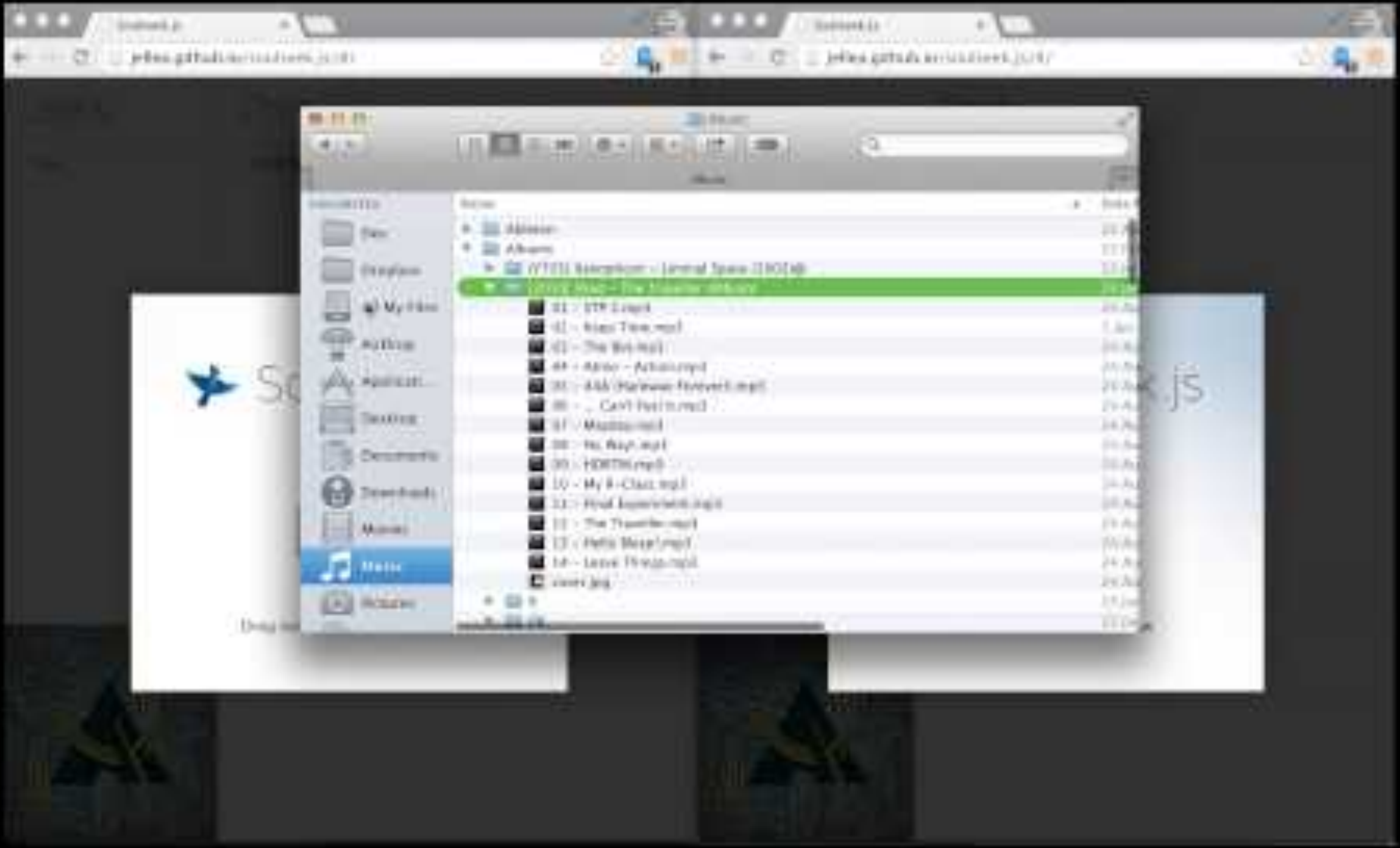
RETWEETS 1,439 FAVORITES 1,248

introducing:
the horror called state

2012: soulseekjs

a p2p filesharing web app.

angularjs, yuk!


Albums
AirDrop
Desktop
Documents
Downloads
Movies
Music
Pictures

```
+------------------------+
|                        |
|  Services & Factories  <------------------+
|                        |                  |
+------------^-----------+                  |
             |                              |
       +-----v----+                 +-------+-------+
       |          |                 |               |
       |  Models  <----------------->  Controllers  |
       |          |                 |               |
       +-----+----+                 +---^---^-------+
             |                          |   |
             |                          |   |
             |  +-----------------------+   |
             |  |                           |
             |  |                           |
     +-------v--v--+                +-------v------+
     |             |                |              |
     |  Templates  <---------------->  Directives  |
     |             |                |              |
     +-------------+                +--------------+
```

wouldn't it be great if...

```
+--------------------+        +--------------------+
|                    |        |                    |
|                    |        |                    |
|       state        +-------->      render       |
|                    |        |                    |
|                    |        |                    |
+--------------------+        +--------------------+
```

2014: soulseekcljs

clojurescript: om, yay!

free goodies

- hotswap code
- time travelling
- introspection
- portable state

hotswap code

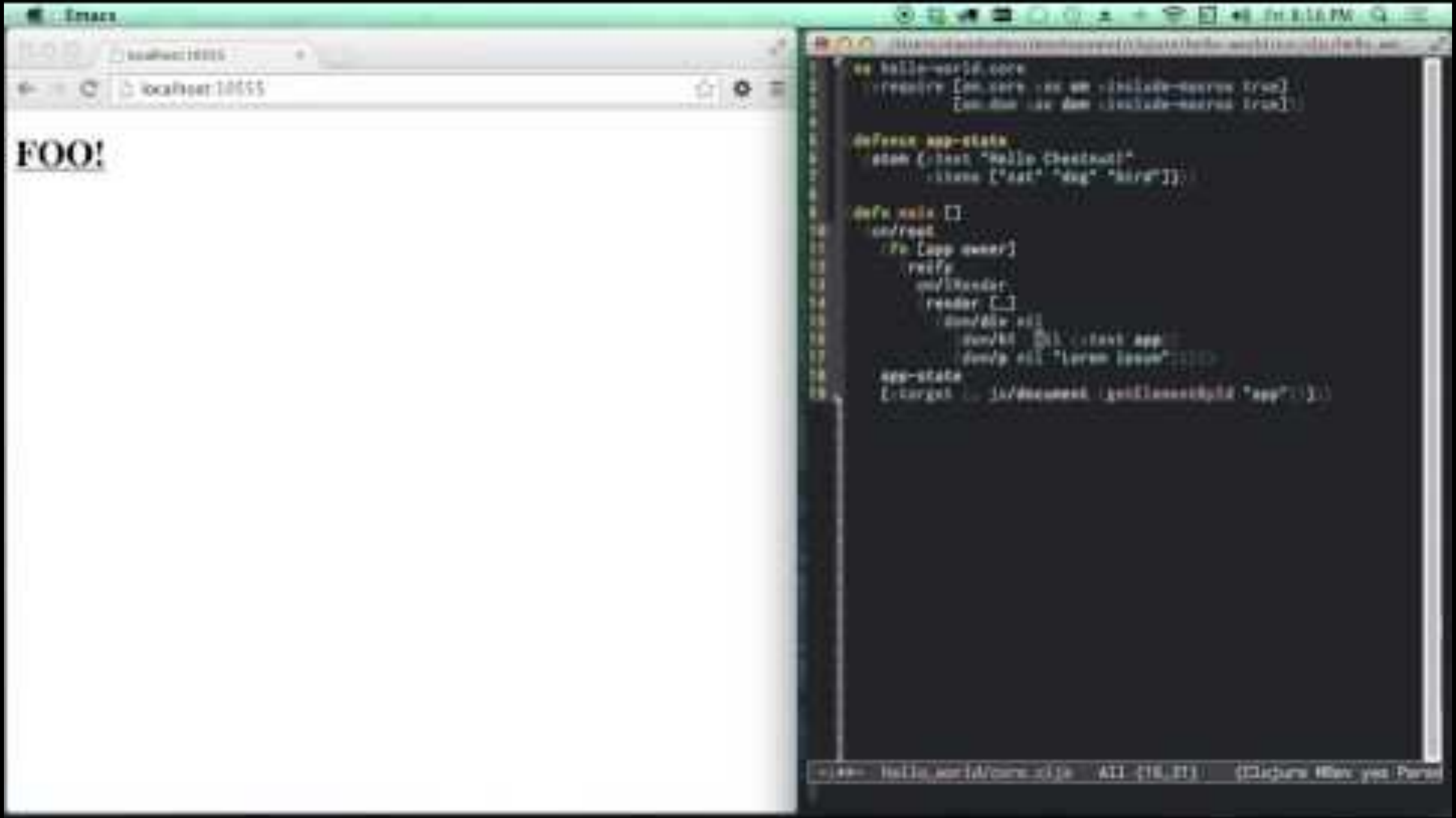
being able to manipulate and rerender.

demo: chestnut by Arne Brasseur[1] or tenzing by Martin Klepsch

demo: quilfiddle by me[2]

[1] Chestnut Video Chestnut repo — Tenzing Repo

[2] QuilFiddle Repo



Emacs
localhost:10555
localhost:10555

FOO!

```
(ns hello-world.core
  (:require [om.core :as om :include-macros true]
            [om.dom :as dom :include-macros true]))

(defonce app-state
  (atom {:text "Hello Chestnut!"
         :items ["cat" "dog" "bird"]}))

(defn main []
  (om/root
    (fn [app owner]
      (reify
        om/IRender
        (render [_]
          (dom/div nil
            (dom/h1 nil (:text app))
            (dom/p nil "Lorem ipsum")))))
    app-state
    {:target (. js/document (getElementById "app"))}))
```

-:**- hello_world/core.cljs   All (16,37)   (Clojure Rev yas Paredit

time travelling

being able to traverse back and forth in time aka. undo/redo.

demo: om todomvc by David Nolen[3]

demo: goya by Jack Schaedler[4]

[3] Om TodoMVC App Undo src

[4] Goya App Repo

```clojure
(def app-history (atom [@app-state]))

(add-watch app-state :history
  (fn [_ _ _ n]
    (when-not (= (last @app-history) n)
      (swap! app-history conj n))
    (set! (.-innerHTML (.getElementById js/document "message"))
      (let [c (count @app-history)]
        (str c " Saved " (pluralize c "State"))))))

(aset js/window "undo"
  (fn [e]
    (when (> (count @app-history) 1)
      (swap! app-history pop)
      (reset! app-state (last @app-history)))))
```

source: http://swannodette.github.io/2013/12/31/time-travel/

introspection

see what data runs under the hood/gui.

demo: tetris by Shaun Le Bron[5]
demo: devcards by Bruce Hauman[6]

[5] Tetris Slides

[6] Devcards Blogpost

portable state

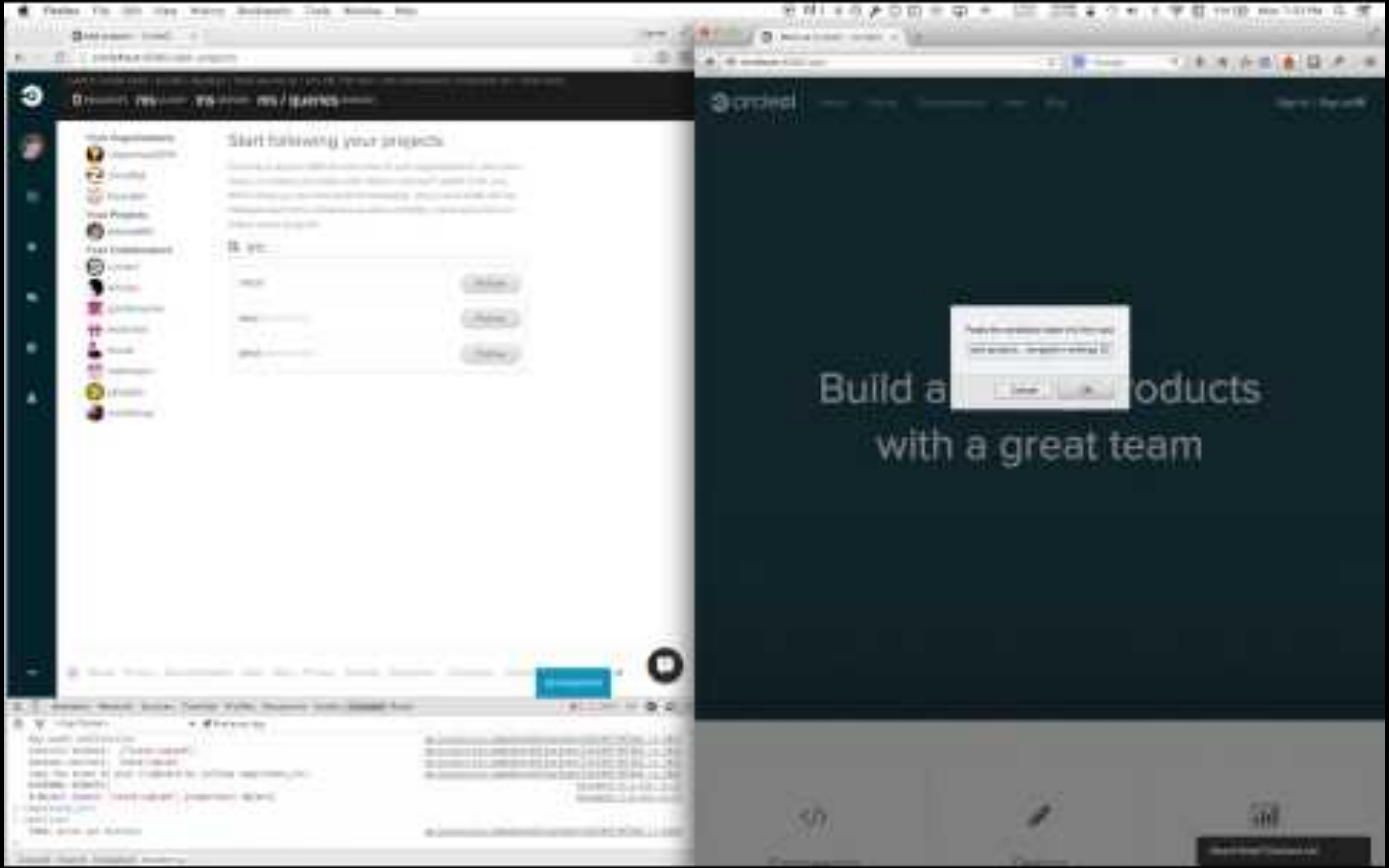
being able to copy paste state.

demo: spagetthi by me[7]

demo: circleci by Circle CI[8]

[7] spaggethi src

[8] circle ci video


circleci
Start following your projects
Build a
roducts
with a great team

```
(defn focus-input [e app owner]
  (if (and (or (.-ctrlKey e) (.-metaKey e)) (not= (.-keyCode e) 86))
    (let [node (om/get-node owner)]
      (set! (.-value node) (t/write w @app))
      (.select node))))

(defn paste-state [app owner]
  (let [input-data (.-value (om/get-node owner))]
     (om/update! app (t/read r input-data))))

(defcomponent clipboard [app owner]
  (did-mount [_]
    (.listen goog/events js/document "keydown" #(focus-input % app owner)))
  (render [_]
          (html [:input {:type "textarea" :style {:opacity 0}
                         :onPaste (fn [] (js/setTimeout #(paste-state app owner) 30))}])))
```

source: https://github.com/jellea/spaghetti/blob/master/src/cljs/spaghetti/core.cljs#L27-L42

other advantages

- compiler
- simple syntax
- core.async
- static typing

om + chestnut (lein)

⬇

reagent + tenzing (boot)

Conclusion:

Want less headaches? Go Clojurescript!

questions? thanks.
twitter + github: @jellea
mail: m@jelle.io