

clijs + 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



Follow

guys i just invented the best programming language



```
2
3
4  Machine,
5
6  Pls make website,
7
8  all responsive like,
9  w/ BIG pictures ooo,
10 use my fav fonts,
11 also fancy menus with whoosh on,
12 load fast pls
13
14 Thanks,
15 Human
16
17 PS no bugs :)
18
19
```

RETWEETS

1,439

FAVORITES

1,248

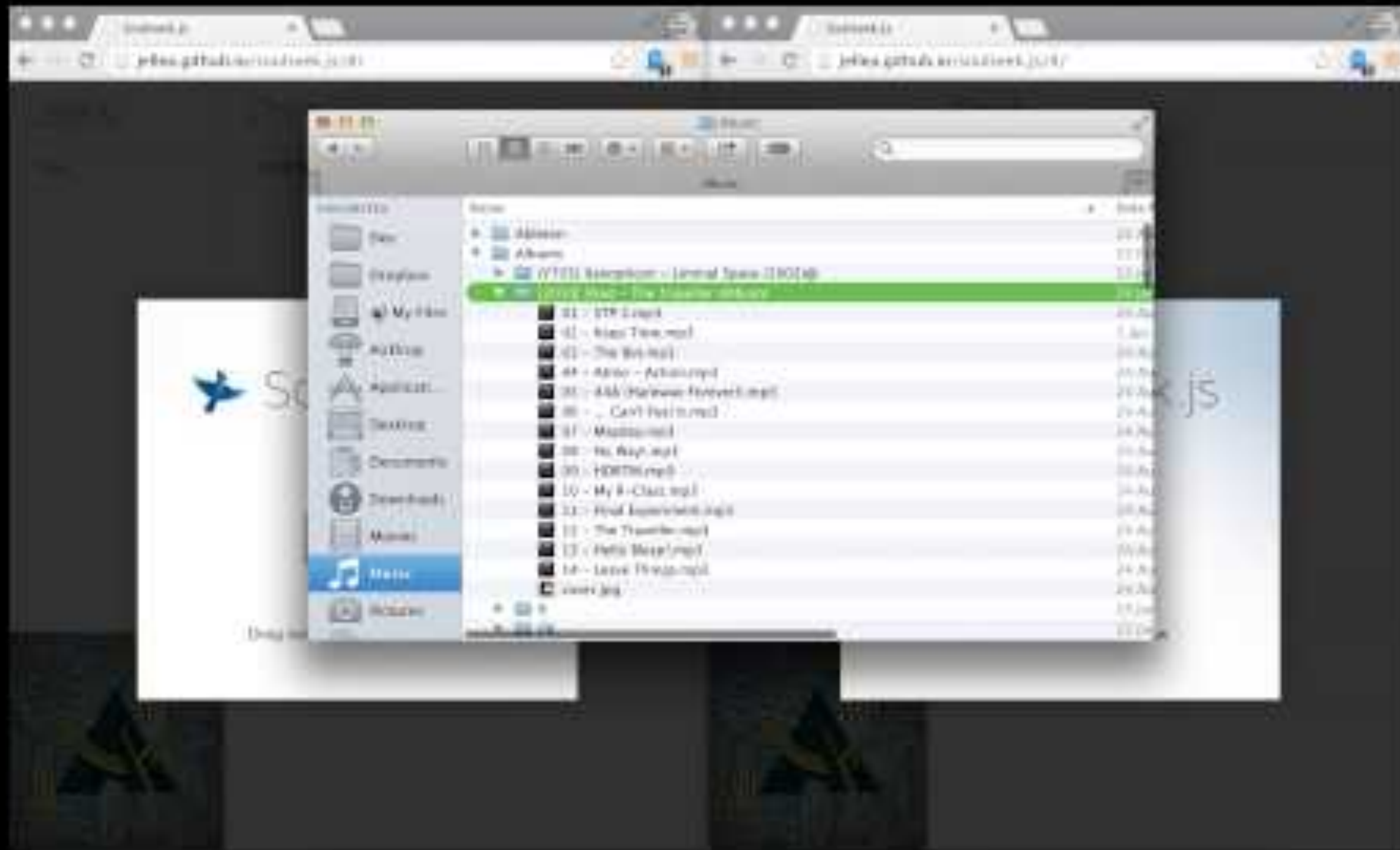


introducing: the horror called state

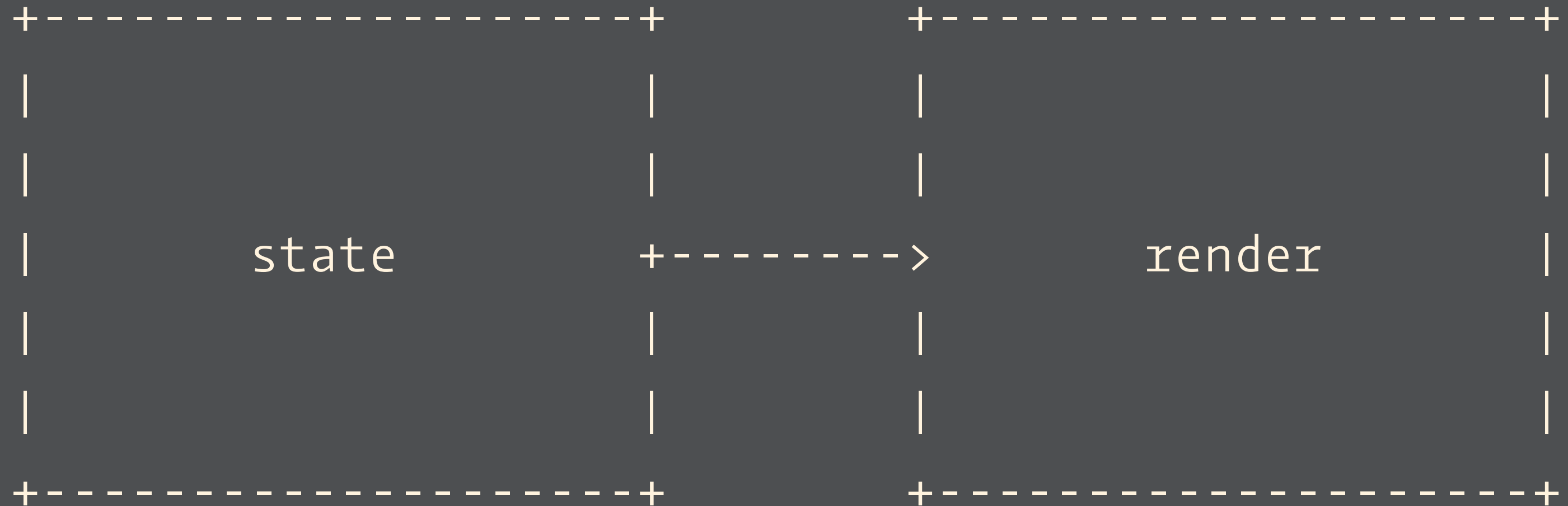
2012: soulseekjs

a p2p filesharing web app.

angularjs, yuk!



wouldn't it be great if...



2014: soulseekcljs

clojurescript: om, yay!

Что это!? Clojurescript!?

- Clojure which compiles to heavily optimised javascript.
- A functional programming language
- (((((a lot of parentheses))))))

free goodies

- hotswap code
- time travelling
- introspection
- portable state
- cross session persistent state

hotswap code

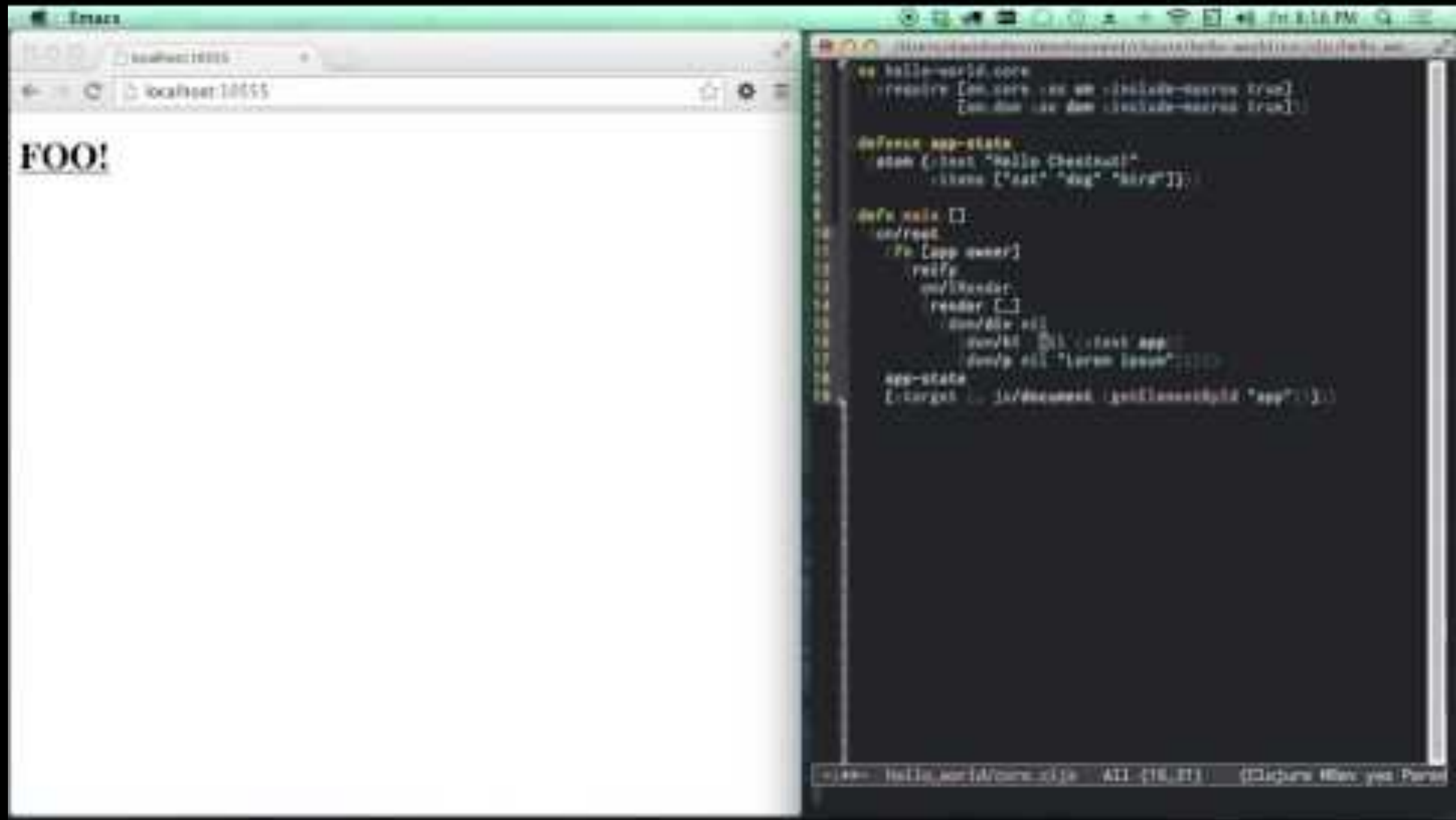
being able to manipulate and
rerender.

demo: chestnut **by Arne Brasseur¹** or tenzing **by Martin Klepsch**

demo: quilfiddle **by me²**

¹ Chestnut [Video](#) [Chestnut repo](#) — Tenzing [Repo](#)

² QuilFiddle [Repo](#)



time travelling

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

demo: om todomvc by David Nolen³

demo: goya by Jack Schaedler⁴

³ Om TodoMVC App Undo src

⁴ Goya App Repo

```
(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 Bron⁵

demo: devcards by Bruce Hauman⁶

⁵ Tetris [Slides](#)

⁶ Devcards [Blogpost](#)

portable state

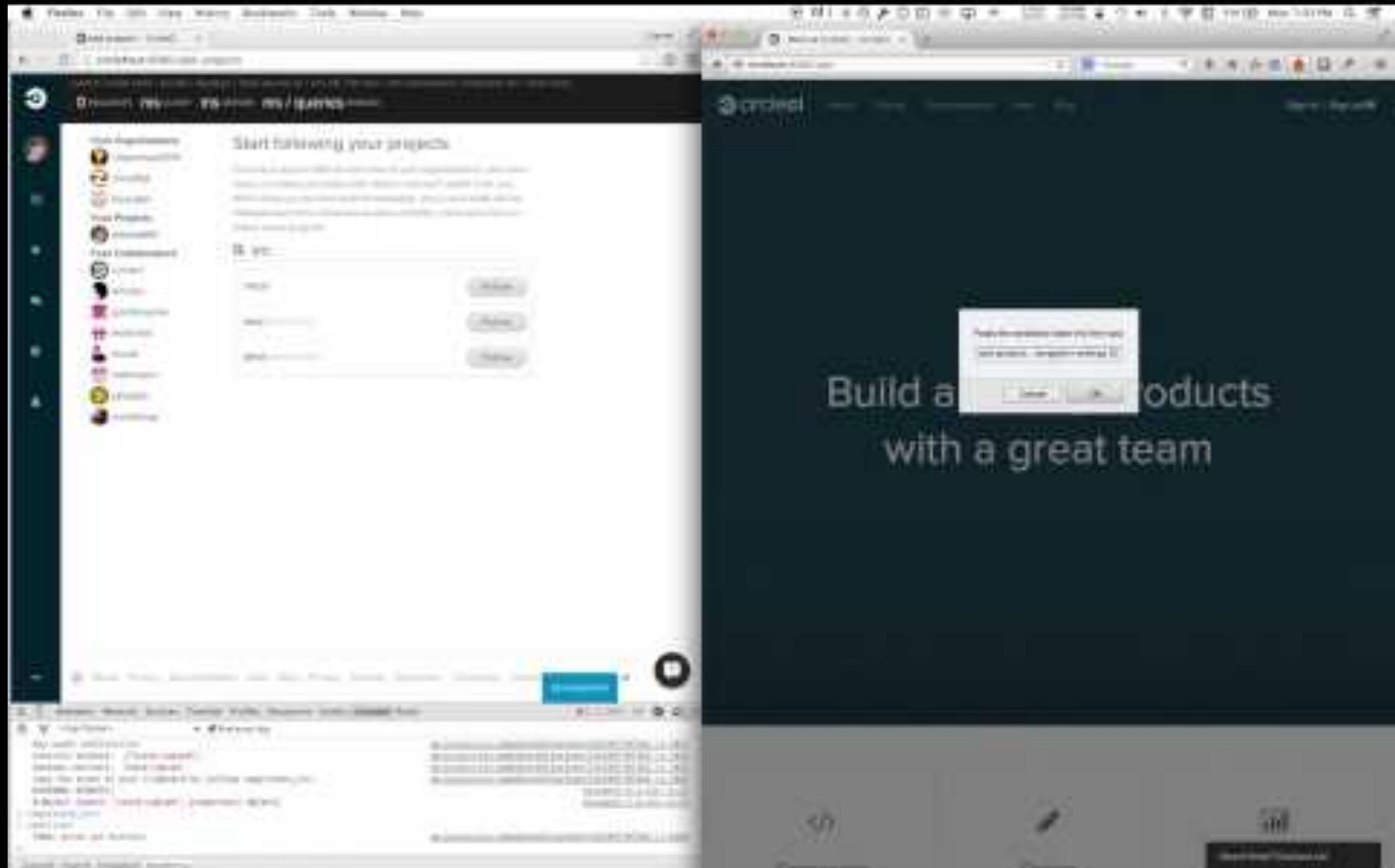
being able to copy paste state.

demo: spaghetti **by** me⁷

demo: circleci **by** Circle CI⁸

⁷ spaghetti src

⁸ circle ci video



```

(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>

cross session persistent state

a.k.a offline first

lib: storage-atom by Alan Dipert⁹

⁹ storage-atom [repo](#)

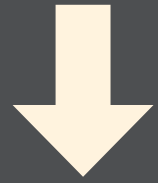
1 LOC

```
(def preferences (local-storage (atom {})) :prefs))
```

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