cljs-info

Learning to Respect ClojureScript's Split-Personality

Frank Siebenlist

frank.siebenlist@gmail.com
https://github.com/franks42/

OUTLINE

- Single-pass ClojureScript compilation
- CLJS REPL
- CLJS Split-Personality
- Where is meta? Where is ns-map?
- Reimplement doc, ns-* function for cljs in clj
- 2-REPL Setup: CLJ-REPL & CLJS-REPL
- Single CLJS-REPL Setup
- Interim One-and-a-half REPL Setup
- cljs-info.repl/cljs->repl
- cljs-ns-browser
- Conclusion
- One more thing...

Single-pass ClojureScript compilation

- ... not a very “Clojuresque” dev process
    - smells like legacy languages

CLJS-REPL

- much better
- almost feels familiar
- ...but where are my cljs introspection fns?
    - no doc, no apropos, no ns-publics, no ns-*

CLJS Split-Personality

- MetaData on CLJ-side
  - no js-vm state awareness
- Live App State on JS-VM side
  - no cljs metadata awareness
- Eval'ed CLJS in REPL can show js-vm state
  - ...but CLJS-metadata not readily available

Where is meta? Where is ns-map?

- CLJS maintains metadata differently from CLJ
  - CLJS has no clj-vars, no clj-ns... no meta {}
    - just clj-symbols mapped to native js-objects
  - CLJS has the @cljs.analyzer/namespaces metadata map
  - CLJS compiler doesn’t share compilation env with runtime environment

Reimplement doc, ns-* function for cljs in clj

- Because metadata is stored differently, doc & ns-* have to be reimplemented
- doc & ns-* implementations have to run in compilation env where metadata is maintained
- cljs-info provides reimplementation of doc & ns-* as clj-fns running on CLJS' compiler-jvm

2-REPL Setup: CLJ-REPL & CLJS-REPL

- CLJS-REPL to compile&eval cljs-forms interactively
- CLJ-REPL to introspect the CLJS metadata thru cljs-doc & cljs-ns-*
- Not ideal...
  ... but it maps to the right mental picture

Single CLJS-REPL Setup

- Let CLJS communicate with CLJ-Server to obtain introspection-data
- Metadata still on CLJ jvm-server
- Use cljs-proxies that make (transparent) rpc-like calls to get doc & ns-* info
- ...or even bring metadata map over to js-vm
  - transparently sync with clj's map
  - (next step... first get proxies working...)

Interim one-and-a-half REPL setup

- Single REPL mode not implemented “yet”
    - “Correct” proxy implementation coming soon...
- Alternative temporary hack available
    - gives illusion of proxies
    - but no real cljs-fns... don’t use them like fns!
- cljs-info provides this ... but as soon as proxies work, you should burn it and forget you ever used it...

cljs->repl & js->repl

- cljs-info also has functions to send cljs-forms programmatically to the cljs-repl
    - cljs->repl
- Great for cljs-test-script written in clj
- Great for programmatically introspecting js-vm state
- There is also a js->repl to send strings of js-code for eval on the js-vm thru the cljs-repl

cljs-ns-browser

- A cljs-ns-browser would be nice
  - browse thru cljs namespaces, functions, vars to look at docstrings, source-code, values
- The cljs-ns-browser relies on introspection facilities like doc, source, ns-*
  - cljs-info provides building blocks
  - note that cljs->repl allows for state introspection, i.e. shows real-time live var's values in js-vm
- Got some of it working... not enough to share yet...
  - but enough to demo

CONCLUSION

- ClojureScript is pretty darn nice
  - ...can you believe people still code in javascript? ;-)
- ClojureScript still a little raw and unfinished
  - Nolan & Company are doing an amazing job
- ClojureScript Dev process has to become more polished
  - Tools are moving slowly...
  - It's still RTFC instead of RTFM...
  - Lein-cljsbuild only gets you half-way there...
  - Emacs... is emacs..., but how about Eclipse? IntelliJ?
- Hopefully there will be enough momentum to take CLJS mainstream
  - requires quite a number of man-years still...

One More Thing...

```
(cljs-info.repl/cljs->repl
    (js/alert "Thanks for listening - go and enjoy ClojureScript!"))
```