A Brief Introduction to Immutant; Sublime Text for Clojure

Steven Byrnes[*]

January 24, 2013

Immutant is an application server for Clojure. Based on JBoss AS7, it provides a rich foundation for building non-trivial applications, including messaging, caching, jobs, clustering, and more. Immutant integrates with Leiningen. It is a sibling project to TorqueBox, and can be configured alongside TorqueBox in the same application server. This allows polyglot applications in Java, Clojure, and Ruby to be written and deployed within the same container.

1 Immutant

You want to deploy . . .

This could be very useful for those who work within an enterprise.

1.1 What

1.2 Who

1.3 Why

If you've worked with Tomcat or other containers, you'll probably feel right at home.

Many enterprises are used to deploying this kind of thing and there's a pretty good chance they're already running Tomcat or JBoss.

~~Wife~~ Boss acceptance factor.

~~Boss~~ Ops acceptance factor.

RedHat!

[*] erewhon@flatland.org

1.4 Demo - A Quick Start

```
# Add {:plugins [[lein-immutant "0.14.1"]}
$EDITOR ~/.lein/profiles.clj
lein immutant install 0.8.0
lein immutant run
# A bunch of output follows...
```

By default, it starts up messaging, scheduling, caching, and a Clojure environment.

It starts up several services, but by default it only listens on localhost, so no worries.

Port 8080 - web server. Port 9990 - JBoss management console.

1.5 Demo - Bringing in the web site

```
git clone git@github.com:chug/cljhouston.git chug
cd chug
lein immutant init
lein immutant run

# New window
$EDITOR src/immutant/init.clj
lein immutant deploy
open http://localhost:8080/cljhouston/
```

1.6 Demo - Adding a chatroom!?

```
# Add messaging dependency and new
$EDITOR src/immutant/init.clj
$EDITOR src/cljhouston/views/welcome.clj

curl -d message='Hello world!' http://localhost:8080/cljhouston/fritter
curl -d message='Hello world again!' http://localhost:8080/cljhouston/fritter
```

Todo:

- form for submitting
- date and IP of who added it
- bounded queue
- ajax

Seems kind of overkill, since we're just running in a single JVM. Can just use the atom. Or. . .

1.7 Demo - Spool up a cluster!

2 Tooling - Sublime Text and Light Table

Been using Emacs since 1986. Version 16 or 17, perhaps. Don't recall. . . Ctrl sequences pretty embedded in muscle memory, but it's years since I've been where I would consider hardcore. (Used to dump and undump my own Emacs.) So I don't need complete keyboard parity. . .

Tried Textmate. Never used it enough. Tried Eclipse, but had issues with version we were using. IntelliJ: bought it and still use it sometimes. Saw a few people who were using Sublime Text, read a few articles, and decided to give it a try. Now I use it for 95% of code writing, but still use Emacs fairly heavily. (For example: I used Emacs and org mode to write these notes.)

Why have I switched? Emacs is awesomely great, but I'm too impatient. Running beta version (24.x), package managers, and periodically when I updated things, they would break in strange ways. Tired, to a certain extent, of debugging my Emacs config. (Which is about 1600 lines.) I'd have over 1000 files loaded into buffers.

Clojure is actually what caused me to switch. Tried to switch on too many modes, and it started trying to do weird things in my ERC mode.

I got better, more sane defaults. Go to file, project management, good language defaults without tons of fiddling.

Repl

First, install the Sublime Text package manager. Then install SublimeREPL. https://github.com/wuub/SublimeREPL

Docs: http://sublimerepl.readthedocs.org/en/latest/

Supports Clojure, plus a few other languages. (CoffeeScript, F#, Groovy, Haskell, Lua, MozRepl, NodeJS, Python, R, Ruby, Scala, shell.)

Created project, turned on side bar, hide projects. So it shows I guess more of a Textmate view. Which is sometimes useful. But %P helps. (Goto anything.)

Things that are different:

%N new file %O open file %S save file %F find

%C / %X / %V %Z - copy cut paste undo

Which aren't Emacs equiv, but standard Mac, and, frankly, mostly easier to hit, so I can deal with it.

Things that I use that are same:

Ctrl-p, Ctrl-n, Ctrl-f, Ctrl-b, Ctrl-a, Ctrl-e, Ctrl-d, Ctrl-o, Ctrl-k, Ctrl-y

Not sure if this is purely Sublime Text, or a side effect of using some standard widget, but I like it. . . . And it's enough that I've stayed.

Other commands:

%P - goto anything %B - build. F4 next error. %/ - toggle comments C-c a - allign add-on. I like lining up assignments, and this does it so quickly for me... Shift % F - find in files %D - select other occurences

$^{\%}Ptoswitchprojects$

https://speakerdeck.com/caiogondim/sublime-text-2-pro-tips https://speakerdeck.com/kentaro/sublime-text-2-for-emacsers

Things that I think are cool:

- multi-line editing.
- can switch projects without saving changes! (unedited buffers return; seems to be very good about remembering state)

Packages:

- align
- ctags

Todo:

- show Lein runs
- show tests
- run stuff in Immutant? deploy to it?
- polyglot programming: slim templates, coffeescript, etc?

Testing frameworks: who is using what?

Testing:

- clojure.test
- Midje
- expectations - https://github.com/jaycfields/expectations, http://blog.jayfields.com/2011/11/clojure-expectations-introduction.html, http://jakemccrary.com/blog/2011/12/16/continuous-testing-with-clojure-and-expectations/
- ClojureCheck - https://bitbucket.org/kotarak/clojurecheck
- LazyTest - https://github.com/stuartsierra/lazytest
- Circumspec - https://github.com/stuarthalloway/circumspec
- speclj - http://speclj.com/