

Configure Clojure(Script) apps with juxt/aero

June, 2017

meet aero

- open-source library for explicit configuration for Clojure/ClojureScript apps provided by **JUXT**
- ~30k downloads from clojars.org ^[1]
- built-in integration with Stuart Sierra's component library ^[2]

similar libs

- cprop ^[3] ~ 49k users
- yohthos/config ^[4] ~ 13k users
- levand/immuconf ^[5] ~ 1.6k users
- grammarly/omniconf ^[6] ~ 1.5 users
- mccraigmccraig/clonfig ^[7] - no clojars account

aero's goals

- provide an easy way to handle explicit intentful configs
- group all the configs in the same file (for all the environments)
- store the configs in repositories but hide the sensitive info
- avoid configure environment variables (why unsafe?)

“aero is edn really, ok?”

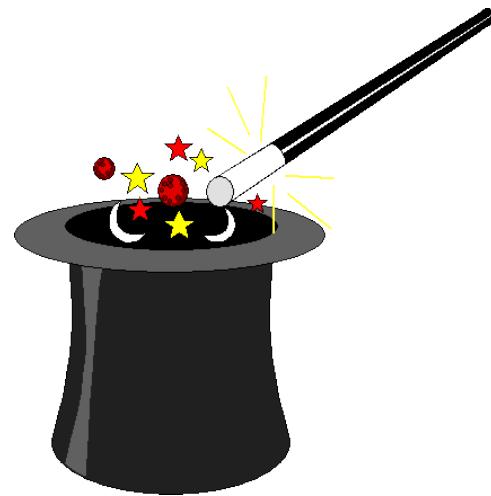
- used by Clojure/Datomic to represent programs/data transfer
- specs ^[8] defined by Rich Hickey
- a set of definitions of acceptable elements
- not a typed system and has no schema
- edn implementations (Haskell, Java, Golang, etc) ^[9]

... more about edn

- UTF-8 encoded chars separated by whitespace or commas
- Strings, chars, symbols, keywords, integers, floating points, collections (lists, maps, vectors, sets), nil
- # - reserved token; expects a reader function which describes behavior
 - eg `#{"a" "set"}`

aero's secret?

Hashtag magic



very basic example

- **define config.edn file:**

```
{:mailbox  
  {:hostname "smtp.gmail.com"  
   :username "joe.doe@gmail.com"  
   :password "DummyPassword"}}
```

- **run in REPL:**

```
user> (aero.core/read-config "config.edn")
```


very basic example

- **define config.edn file:**

```
{:mailbox  
  {:hostname "smtp.gmail.com"  
   :username "joe.doe@gmail.com"  
   :password "DummyPassword"}}
```

- **run in REPL:**

```
user> (aero.core/read-config "config.edn")
```

```
{:mailbox {:hostname "smtp.gmail.com", :username  
"joe.doe@gmail.com", :password "DummyPassword"}}
```

another basic example

- **enhance config.edn file:**

```
{:mailbox
  {:hostname #profile {:prod "smtp.companydomain.com"
                        :dev  "smtp.gmail.com"}
   :username #profile {:prod "office@companydomain.com"
                        :dev  "joe.doe@gmail.com"}
   :password #profile {:prod "CompPass4Joe@2017"
                        :dev  "DummyPassword"}}
```

another basic example

- **run in REPL:**

```
user> (aero.core/read-config "config.edn"  
{:profile :prod})
```

another basic example

- **run in REPL:**

```
user> (aero.core/read-config "config.edn"  
{:profile :prod})
```

```
{:mailbox {:hostname  
"smtp.companydomain.com", :username  
"office@companydomain.com", :password  
"CompPass4Joe@2017"}}
```

#profile

- **for the hostname element in EDN:**

```
:hostname #profile {:prod "smtp.companydomain.com"  
                   :dev  "smtp.gmail.com" }
```

#profile

- **for the hostname element in EDN:**

```
:hostname #profile {:prod "smtp.companydomain.com"  
                  :dev  "smtp.gmail.com" }
```

- **check aero.core associated reader function:**

```
(defmethod reader 'profile  
  [{:keys [profile]} tag value]  
  (cond (contains? value profile) (get value profile)  
        (contains? value :default) (get value :default)  
        :otherwise nil))
```

hidden pass example

- **define .secrets.edn file in HOME directory:**

```
{ :mailbox-password-prod "CompPass4Joe@2017" }
```

Note: secrets.edn file should be present in the HOME folder of any user that runs the app (ie including on the prod machine)

hidden pass example

- **enhance config.edn file:**

```
{:secrets #include #join [#env HOME "/.secrets.edn"]
 :mailbox
  {:hostname #profile {:prod "smtp.companydomain.com"
                       :dev  "smtp.gmail.com"}
   :username #profile {:prod "office@companydomain.com"
                       :dev  "joe.doe@gmail.com"}
   :password #profile {:prod ^:ref [:secrets :mailbox-
password-prod]
                       :dev  "DummyPassword"}}
```


hidden pass example

- **run in REPL:**

```
user> (aero.core/read-config "config.edn"  
{:profile :prod})
```

hidden pass example

- **run in REPL:**

```
user> (aero.core/read-config "config.edn"  
{:profile :prod})
```

```
{:secrets {:mailbox-password-prod  
"CompPass4Joe@2017"}, :mailbox{:hostname  
"smtp.companydomain.com", :username  
"office@companydomain.com", :password  
"CompPass4Joe@2017"}}
```

dig into #env, #join, #include

```
:secrets #include #join [#env HOME "/.secrets.edn"]
```

- **#env** HOME <=> (System/getenv "HOME")

```
user> (System/getenv "HOME")
```

```
"/home/kitkat"
```

dig into #env, #join, #include

```
:secrets #include #join [#env HOME "/.secrets.edn"]
```

- **#env** HOME <=> (System/getenv "HOME")

```
user> (System/getenv "HOME")  
"/home/kitkat"
```

- **#join** coll <=> (apply str coll)

```
user> (apply str [(System/getenv "HOME") "/.secrets.edn"])  
"/home/kitkat/.secrets.edn"
```

dig into #env, #join, #include

```
:secrets #include #join [#env HOME "/.secrets.edn"]
```

- **#env** HOME <=> (System/getenv "HOME")

```
user> (System/getenv "HOME")  
"/home/kitkat"
```

- **#join** coll <=> (apply str coll)

```
user> (apply str [(System/getenv "HOME") "/.secrets.edn"])  
"/home/kitkat/.secrets.edn"
```

- **#include** filepath <=> apply path resolver and file reader:

```
:secrets {:mailbox-password-prod "CompPass4Joe@2017"}
```

extend tag magic – define #myport

- create new EDN file:

```
{:port #long #myport :favorite}
```

extend tag magic – define #myport

- **create new EDN file:**

```
{:port #long #myport :favorite}
```

- **define the function associated with the new tag:**

```
(defmethod reader 'myport  
  [{:keys [profile] :as opts} tag value]  
  (if (= value :favorite) 22 7022))
```

extend tag magic – define #myport

- **create new EDN file:**

```
{:port #long #myport :favorite}
```

- **define the function associated with the new tag:**

```
(defmethod reader 'myport  
  [{:keys [profile] :as opts} tag value]  
  (if (= value :favorite) 22 7022))
```

- **run in REPL:**

```
user> (:port (aero.core/read-config "config.edn"))
```


extend tag magic – define #myport

- **create new EDN file:**

```
{:port #long #myport :favorite}
```

- **define the function associated with the new tag:**

```
(defmethod reader 'myport  
  [{:keys [profile] :as opts} tag value]  
  (if (= value :favorite) 22 7022))
```

- **run in REPL:**

```
user> (:port (aero.core/read-config "config.edn"))
```

```
22
```

extend tag magic – define #myport

- **change in EDN file only (reader function remains the same)**

```
{:port #long #myport :not-favorite}
```

extend tag magic – define #myport

- **change in EDN file only (reader function remains the same)**

```
{:port #long #myport :not-favorite}
```

- **re-run in REPL:**

```
user> (:port (aero.core/read-config  
"config.edn"))
```

extend tag magic – define #myport

- **change in EDN file only (reader function remains the same)**

```
{:port #long #myport :not-favorite}
```

- **re-run in REPL:**

```
user> (:port (aero.core/read-config  
"config.edn"))
```

```
7022
```

links

- [1] <https://clojars.org/aero>
- [2] <https://github.com/stuartsierra/component>
- [3] <https://clojars.org/cprop>
- [4] <https://clojars.org/yogthos/config>
- [5] <https://clojars.org/levand/immuconf>
- [6] <https://clojars.org/com.grammarly/omniconf>
- [7] <https://github.com/mccraigmccraig/clonfig>
- [8] <https://github.com/edn-format/edn>
- [9] <https://github.com/edn-format/edn/wiki/Implementations>