U053032QC : `1.1.0-SNAPSHOT` means "the version that will eventually be released as `1.1.0`"
U053032QC : i.e., `*-SNAPSHOT` comes before `*`
U053032QC : under the hood, every time you `lein install` a snapshot version, a timestamped artifact is added to your maven repo, and the SNAPSHOT name just points to the latest of these
U053032QC : if you use the `lein-release` plugin then when you're ready to release it, you run `lein release` and it'll transition your `1.1.0-SNAPSHOT` to `1.1.0`, commit, tag and deploy that to your maven repo, and then bump the version to `1.1.1-SNAPSHOT`, so you're ready to start working on the future `1.1.1` release.
U6AKZKQVC : <@U053032QC> Thanks!
U0ALQHJRF : <http://github.com/degree9/meta|github.com/degree9/meta> has a fully open ended UI solution
U0ALQHJRF : uses edn and a template to generate a default UI or you can provide an alternate template or final cljs file for the ui
U5Z4ECHCM : Hate to interrupt, but I'm having trouble splitting up the middleware for my different routes in Compojure
U5Z4ECHCM : Anyone have information / a blog post / etc on the topic?
U46LFMYTD : I have a question about structural sharing in nested maps. Suppose I have a map called mymap that looks like this```
mymap
=> {:mykey [{:A 1, :B 2} {:A 2, :B 2} {:A 3, :B 2}]}
```
I want to write a function that goes down and increments each value for :A
I can do this with
```
(assoc mymap :mykey (map #(assoc % :A (inc (:A %))) (:mykey mymap)))
{:mykey ({:A 2, :B 2} {:A 3, :B 2} {:A 4, :B 2})}
```
do the keys :B "point to the same place in memory" after this modification as the original maps? That is, is there structural sharing going on between the maps in the vector before and after this modification?

U0CHY4VNW : Asymmetry in `string/split` and `string/join` makes me sad : ( ```  parts (string/split kafka #":")
host (string/join ":" (butlast parts))
```

U5NAUMCAD : Hi all, I am trying to write some test in clojure and I need to do some Mocks and proxies, however I cannot figure out how to proxy some java objects and methods
I have this function

U5NAUMCAD : ``` (defn singletest [cert]
  (let [x500principal (.getSubjectX500Principal cert)
        dn (.getName x500principal)]
    (apply str "abcdefg"))) ```

U5NAUMCAD : cert is a java object of type X509CertImpl
U5NAUMCAD : I know that I am not using x500principal and dn; but it is an example
U5NAUMCAD : However I am having a hard time trying to do the test
U5NAUMCAD : ``` (deftest singletest (let [cert (.X509CertImpl)
        x500principal (proxy [X509CertImpl] []
                        (getSubjectX500Principal [] nil))
        dn (proxy [X500Principal] []
                        (getName [] "works"))]
      (is (= "" (c/singletest cert))))) ```

U5NAUMCAD : throws me java.lang.IllegalArgumentException: Malformed member expression, expecting (.member target ...)
U5NAUMCAD : any idea??
U5NAUMCAD : someone has worked with certificates before :sob::sob:
U050MP39D : what is c?
U050MP39D : c/singletest looks like a syntax error