

U060FKQPN : clojure implements its basic operations in terms of interfaces not protocols

U11BV7MTK : here's how it was done in the priority-map
https://github.com/clojure/data.priority-map/blob/master/src/main/clojure/clojure/data/priority_map.clj#L255

U060FKQPN : right, you can do that if you're in control of new types but it's simply not possible to retrofit on existing types like `java.util.HashMap`

U060FKQPN : (and even if it could be possible, `conj` is defined as an operation on persistent colls, so on mutable ones it would be breaking its contract)

U0NBGRGD6 : okay, so if I want to wrap something, for example `Array[ResultSet]` to vector of map. I can't just extend-type, I should iterate it.

U0NBGRGD6 : thanks for help!

U051SS2EU : <@U0NBGRGD6> you could define a multimethod which is identity for vector of hashmap, but uses `into` to populate clojure datatypes if it gets arrays and java.util.HashMap

U5NAUMCAD : Hi all, I am trying to load a keystore file in clojure:

U5NAUMCAD : (let [a (try (java.io.FileInputStream. keyObjectFile) (catch Exception e (println (str "1caught exception: " (.getMessage e))))))
 (println a)
 (try (.load keystore a) (catch Exception e (println (str "2caught exception: " (.getMessage e))))))

U5NAUMCAD : but I am getting: java.io.FileInputStream cannot be cast to java.security.KeyStore\$LoadStoreParameter

U0509NKGK : <@U5NAUMCAD>
 ...
 (defn load-key-store [keystore-filename keystore-password]
 (with-open [is (<http://clojure.java.io/input-stream|clojure.java.io/input-stream> keystore-filename)]
 (doto (java.security.KeyStore/getInstance "JKS")
 (.load is (.toCharArray keystore-password))))
 ...

U0509NKGK : works great :+1:

U5NAUMCAD : mm i do not have the keystore-password

U5NAUMCAD : i need only the file

U0509NKGK : check the javadoc of `java.security.KeyStore/getInstance` maybe there's a method that doesn't need a password?

U5NAUMCAD : actually, the load does not need a password

U0JFCEH9P : the password can be null i.e. nil

U5NAUMCAD : can I set something like:

U5NAUMCAD : ```(defn load-key-store [keystore-filename] (with-open [is
 (<http://clojure.java.io/input-stream|clojure.java.io/input-stream> keystore-filename)]
 (doto (java.security.KeyStore/getInstance "JKS")
 (.load nil))))```

U5NAUMCAD : ????

U0JFCEH9P : sure

U0JFCEH9P : or you could use the `defn` from above and just call `(load-key-store filename nil)`

U0JFCEH9P : then you won't have to change it when you *do* have a password on the keystore

U5NAUMCAD : ok perfect I am gonna try and let you know what happens!!!

U5NAUMCAD : mmm now I have a problem when I put in the jetty configuration the Keystore object it send me this error in the browser

U5NAUMCAD : SSL_ERROR_NO_CYPHER_OVERLAP

U5NAUMCAD : in jetty I put

U5NAUMCAD : ```:keystore (load-key-store "filename")```