

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") ```

U5NAUMCAD : and send me that message; but if I put the path it works,.

U5NAUMCAD : ```:keystore "path to file" ```

U5NAUMCAD : any idea why this is happening?

U5NAUMCAD : :cold_sweat:

U0NCTKEV8 : it means the ssl server and your browser are unable to agree on a cipher suite

U0NCTKEV8 : which is likely due to using an out dated version of either

U0JFCEH9P : BTW I would strong recommend *against* doing SSL termination in Jetty

U0JFCEH9P : it's much better to do it with a reverse-proxying load balancer

U0JFCEH9P : e.g. NGINX or an ELB in AWS

U0JFCEH9P : you can do SSL termination in Jetty just fine but it just doesn't seem worth the effort when other tools do it better

U0JFCEH9P : if you really really really need one host to serve up HTTPS from a single Java process, then sure