

#CLOJURE.cl CODES AGAINST AN ANTI-NSA
NETWORK WHICH CLAIMED TO BUT LEGAL
RIGHTS AND CASE DOCKETS AGAINST THE NSA
TECHNOLOGIES DEVELOPT INCLUDING COMM
SIGNALS IN THE EASTERN AND WESTERN
STATES U.S.A. AND ALSO THE PRESENCE OF
NSA IN SOUTHWESTERN TEXAS AND FLORIDA.
MEDIA REPORTED ANTI-NSA NETWORK AGAINST
THE COMM SIGNALS OF NSA EMPLOYEE
COMPUTER SYSTEMS AND MAINFRAMES. BAD
COMPUTER RUFFIANS WERE ATTEMPTING TO
MAKE ANGULAR.jsn-style WEBLOGS TO AGITATE
NSA NETWORKS AND COMM SIGNALS.

<ns.Multiply.Bug.Fix.html.cl?rubyclojure.load2.utahtexas
.ns.<n> <|>:

```
#CLOJURECOMMENT: THIS defn CREATES  
datashift a6:CONNECTION TO RECONNECT  
NEWhtmlFOR connection allowing connection [a c].
```

```
defn (Webliftern (connection time date)) {[a b c d  
n=(n-1)+utah.html.ns<|> ]} connection connection.cl
```

```
  If connection con {[a b c d ]}  
http://www.ryder.com/en/leasing.aspx
```

```
    then datashift :: a ** 6 | connection+ time _b-  
1.cl
```

```
#CLOJURECOMMENT::THISBOOL=TRUE.   Sorts  
[a], [b]  for n</1</3? FOR ANY {[String [a] [b]]}
```

```
  let (Webliftern (Newhtml :: [a b.html-1] ))) (n-  
1).connection == BOOL==TRUE if n </ a ?? c + b
```

```
  otherwise TRUE -= n + a == FALSE if n >/ a
```

```
  sort::[a]=BOOL-1:
```

```
    [b]==BOOL-2::
```

[c]=BOOL-

<http://www.ryder.com/en/leasing.aspx>

\$\$\$ (a) == Webliftern.newhtml.ns.n.??

SENDFILE3: BOOL+2.n<n>.ns.c++b<n>

#CLOJURECOMMENT:::Resendconnection creates
close connection flush for .aspx for HTML.ns.BOOL

<Resendconnection

flush::Newhtml:::nsa.gov.ns.htmlcustomercare.2BOOL=
=TRUE. n + a

>?<n>/:http://www.datamark.net/call-
centers/solutions/by-service/order-fulfilment/c && {[a
b]} .aspx

<close connection time date

BOOL=TRUE.ns.2+\$\$\$ (a)

```
n> ResendWebliftern .html.ns.cl(n flush :
close.ns if
Resendconnection==BOOL=FALSEResend.ns.html<n>
```

```
#CLOJURECOMMENT: THIS letfn(.ns) CREATES A
TIME DATE STAMP FOR close connection a && c
time date
```

```
let (close connection a && c time date )))
browser.ns.nt.BOOL
```

```
http://www.meetup.com/pro/mongodb/#addingstamp<m
ynotes-0>stampprintln( time, date "TimeDate.n".n
```

```
OPEN=BOOL | time(date)stampprintln(time
date, "It is {[time]} and it is { [date ] } "
```

```
flush connection | close [a b c d] list name.cl
```

```
open [Cardgame::Hearts].ns.clplay==BOOL
```

```
otherwise play -= close.ns.OPEN==BOOL
```

Play.hearts[a c]

listname.hearts.clSENDFILE=browserBOOL=TRUE