



```
(defn file-handler [{:keys [queue]} {:keys [^File kind] :as event}]
```

(f/insert-file on {name (getNamel file) :prosed (Date.)})

```
(defn weapons-query-handler [{:status[name]} params :keys [conn] :as request}]
```

{:jdbc/connnection{:connnection-un'jdbc:h2:men:men\_only'}}

```
(defn add-weapons-handler [{:keys [params] :as request}]
```

(timbre/debug(str 'Detected hang to file:'))

```
(let [name & weapons] (map cs/trim (cs/spplit line #', '')))
```



```

::jdbc/init::jdbc/connnection}

```

```
(let [data (map(fn [kv] { :name kv :weapon v }) params)]
```

`hawk/watch { :groups { :paths ["example"] }`

(defmethod dig/init-key *jdbc/init* [\_ {*keys* [conn]}])

*;(tindre/debug'checking for new items in queue.')*

**(when-one [task (dq/take! queue-name 10 nil)])**

```
(defn queue->dsdb [{:keys [queue-name queue-dsdb]}])
```

*is cduling/job* { *job* #'queue->dsdb



`{:data{:precisionreit.it.cerec/cerec`

`is_durable_queue { is_delete-on-halt? true`

(**tinbre/debug** **'Putting data into data script'**)

*is chdule { : in [5 : seconds : every : second }*

*:dsdb(ig/ref::datasrcip/connnection)}*

#!/web/server { :host "0.0.0.0"

*:con (ig/ref::data\$inpt/connetion)*

(d//t ransact! d sdb[(line->read@task)])



**(ok(withoutstr(pp/pprintrequest)))**

```
(defn files-handler [{:keys [sqn] :conn}]
```

;We were able to connect the routes here

(ok(w/weapons@anon(keyparams)))

(cs/ends-with? (.getNane file) "csv"))

**(ok(fatal-procesed-files q1-conn))**

(constantly(not-found'Not found'))

(tinbre/debug "Adding data to queue.")



['/add\_weapon' add\_weapon\_handler])

*:queue (ig/ref::durableref/queue)*

(when (and (#*modify* *read*) kind)

*sqz-conn(ig/ref::jdb/cnnec tion)*

[[ '/weapons' weapons\_query\_handler ]]

*::conn(i::jdbc/connnection)}*

```
["/files":get_files_handler]])
```

*queue(ref::durablerefs)*



(dq/put!queue:my-queue line))

*data/crypt/connections/*schema

(w/everybody's-weapon@com))

(with-open [r(io//reader file)])

*minddleware* [params/wrap-params]

[[['/echo'},{:get\_echo\_handler}]]

[basic-routes-weapons-routes]

(define sys (create conf))



(defnetcher [request])

mind[eware//wrap-format]})

(as doc: *weapon* weapons)

**(doseq[line](line-seq r))**

(define-line->record [line])

*hand/ler* # 'final-hander' ]

(w/weapons@corn[ame])

`:"directory"//tmp"}`



(d/transact!commanddata)

**(dq//complete!task)()**

*#hand/ler* #'handler})

*queue-name:my-queue*

(cond->{ :name name})

(def weapons-routes

Routes - Atlanta

**(ring/ring-handler**



(defbasic-inputs

*;Gzobazhander*

**(f/setup con)**

**(.exits file)**

**(.isFile file file)**

*print: 30000*

(seq weapons)

**(ring/router**



Handletter

defrouter

(defconfio)

*Handzlers*

)

i

f

n

a

m

e

*Router*

not a real

ctx)



(

O

K



do

o



