



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

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

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

{:jdbc/connection{:connection-unix'jdbc:h2:mem:mem\_only'}}

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

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

```

::jdbc/init::jdbc/connnection}

```



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

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

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

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

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

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

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

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



*if \$chcdurl{ing/job}{'queue->dsdb*

`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::datastcpt/connctio)}*

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

*:cn (ig/ref::data\$cnip/cnnet)*

(d//t rand t! d s db[(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-con))**

`(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/c/connec tion)*

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

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

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

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



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

*data/crypt/connections/*schema

*minddleware* [params/wrap-params]

(w/everybody's-weapon@com)

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

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

[basic-routes-weapons-routes]

(define-sys(create-confing))



(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

(defconfio)

defrouter

*Handzlers*

)

i

f

n

a

m

e

*Router*

not a real

ctx)



(

O

K



do

o



