



(tinbre/debug(str 'Detected hang to file: 'getNanefile))

`{:jdbc/cnnec tion{:cnnec tion-un i'jdbc:h2:mem:mem_only'}}`

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

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

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

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

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



*::jdbc/init::jdbc/condition}*

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

(defnet hodb :*ig* / *init* - key :*jdbc* / *init* [ \_ { :*keys* [ conn ] } ]

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

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

`{:data{:precisionreitit.condition.spec/condition`

*;(tibre/debug'checking for new ints in queue.')*

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



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

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

`if du rable/queue { if det-on-halt? true`

webserver { host "0.0.0.0"

*is the  $\{in[5:second]: every: second\}$*

*:dsdb(ig/ref::data\$ip\$connection)*

*:con (ig/ref::data input/connetion)*

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



(d//transact!dsdb[(line->read@task)])

*We were able to connect here*

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

`(ok(without-str(pp/print request)))`

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

['/add\_weapon' add\_weapon-handle'])

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

**(ok(f/all-procesed-files ql-con))**



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

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

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

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

$\{conn(i_g/ref:j_dbc)/connec tion\}$

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

*data/crypt/connections/schema*

[[ '/weapons' weapons-query-handler ]]



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

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

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

*minddleware* [params/wrap-params]

```
[['/echo'{:aget echo-handler}]]
```

[basic-routes weapons-routes]

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

(defn echo-handler [request])



middeware/wrap-format}]})

(define sys(create conf))

(define-line->record [line])

(as doc: *weapon* weapons)

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

`:"directory"//tmp"}`

(w/weapons@conn[name])

**(d/transact!command)**



$(dq/d\text{complete!task})()$

*#hand/zer* #'handler})

queque-nammy-queque

*:hand/ten#':final-e-handten}]*

(cond=>{:name name})

**(ring/ring-handler**

(def weapons-routes

*Routes - Atlanta*



(defbasic-inputs

**(f/setup con)**

*;Gzobazhander*

**(.isFile file file)**

**(.exits file)**

*print* 30000

(seq weapons)

(def handler



**(ring/router**

(defconfi

defrouter

*Handzlers*

)

i

f

n

a

m

e

*Router*

route er

ctx)



(

o

k





