

Clojure-listing i pandoc

Truls Thirud

Eksempelkode

Her kommer en kodeliste med clojure-kode som bør være fargekodet og i en liten font.

```
1 (ns guestbook.core
2   (:require
3     [guestbook.handler :as handler]
4     [guestbook.nrepl :as nrepl]
5     [luminus.http-server :as http]
6     [luminus-migrations.core :as migrations]
7     [guestbook.config :refer [env]]
8     [clojure.tools.cli :refer [parse-opts]]
9     [clojure.tools.logging :as log]
10    [mount.core :as mount])
11   (:gen-class))
12
13 ;; log uncaught exceptions in threads
14 (Thread/setDefaultUncaughtExceptionHandler
15   (reify Thread$UncaughtExceptionHandler
16     (uncaughtException [_ thread ex]
17       (log/error {:what :uncaught-exception
18                   :exception ex
19                   :where (str "Uncaught exception on" (.getName thread))}))))
20
21 (def cli-options
22   [["-p" "--port PORT" "Port number"
23     :parse-fn #(Integer/parseInt %)]])
24
25 (mount/defstate ^{:on-reload :noop} http-server
26   :start
27   (http/start
28     (-> env
29       (assoc :handler (handler/app))
30       (update :port #(or (-> env :options :port) %))
31       (select-keys [:handler :host :port :async?]))))
32   :stop
33   (http/stop http-server))
34
35 (mount/defstate ^{:on-reload :noop} repl-server
36   :start
37   (when (env :nrepl-port)
38     (nrepl/start {:bind (env :nrepl-bind)
39                   :port (env :nrepl-port)}))
40   :stop
41   (when repl-server
42     (nrepl/stop repl-server)))
43
44
45 (defn stop-app []
46   (doseq [component (:stopped (mount/stop))]
47     (log/info component "stopped"))
48   (shutdown-agents))
49
50 (defn start-app [args]
51   (doseq [component (-> args
52                         (parse-opts cli-options)
53                         mount/start-with-args
54                         :started)]
55     (log/info component "started"))
56   (.addShutdownHook (Runtime/getRuntime) (Thread. stop-app)))
57
58 (defn -main [& args]
59   (-> args
60     (parse-opts cli-options)
61     (mount/start-with-args #'guestbook.config/env))
62   (cond
63     (nil? (:database-url env))
64     (do
65       (log/error "Database configuration not found, :database-url environment variable must be set before running")
66       (System/exit 1))
67     (some #{"init"} args)
68     (do
69       (migrations/init (select-keys env [:database-url :init-script]))
70       (System/exit 0))
71     (migrations/migration? args)
72     (do
73       (migrations/migrate args (select-keys env [:database-url]))
74       (System/exit 0))
75     :else
76     (start-app args)))
```