Clojure Web-Application 101

Michael Vitz
innoQ

DEVOXX™

Clojure

- Lisp on the JVM
- Functional
- Dynamically typed
- Immutable data structures
- macros

Data structures

```
3  3.14  3/2                          true   false

      “Hello World"    \a    #“Ch.*se”

foo                :first
                                        (“Hello” :first)

{ :name “Michael”,                          [3 4 3]

  :company “innoQ” }       nil          #{3 4 3}
```

DEVOXX™

Syntax

"You've just seen it"

- Rich Hickey

Functions

```
(+ 1 2)
› 3
(:city {:name “Devoxx”
        :city “Antwerp”})
› “Antwerp”

(map inc [1 2 3])
› (2 3 4)
```

Functions

```
(fn [x y] (+ x y))
```

```
(def add
  (fn [x y] (+ x y)))
```

```
(defn add
  [x y]
  (+ x y))
```

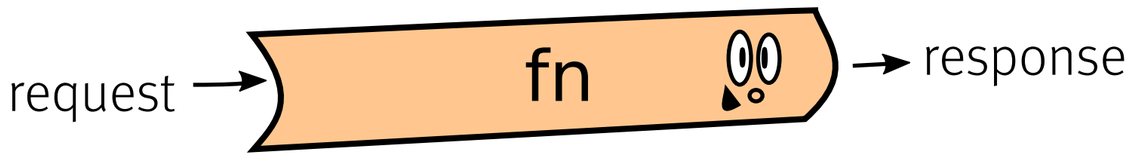
Links

- http://clojure.org
- blog.cognitect.com/blog/2016/1/28/state-of-clojure-2015-survey-results
- https://juxt.pro/radar.html
- http://www.clojure-toolbox.com

DEVOXX™

Web-Applications

Web-Application



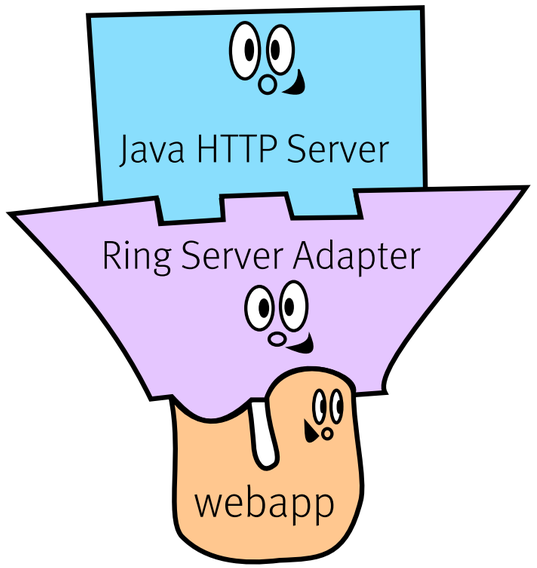
Web-Application

```
(defn greet-controller
  [request]
  {:status 200
   :headers {“Content-Type” “text/plain”}
   :body “Hello, world!”})
```

Ring

https://github.com/ring-clojure/ring

Ring



Ring: Request

```
{:uri “/greet”
  :query-string “name=Michael”
  :request-method :get
  :headers { … }
  …}
```

Ring: Response

```
{:status 200
 :headers { … }
 :body “Hello, Michael!”}
```

Ring: Handler

```
(defn greet-controller
  [request]
  {:status 200
   :headers {“Content-Type” “text/plain”}
   :body “Hello, world!”})
```

Ring: Adapter

- Jetty
- Servlet
- http-kit
- Tomcat
- …

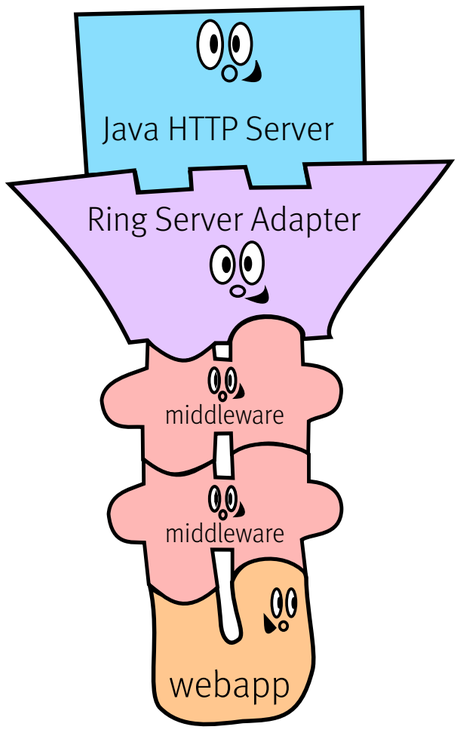
Ring: Middleware

```
(defn wrap-logging
  [handler]
  (fn [request]
    (print request)
    (let [response (handler request)]
      (print response)
      response)))
```

Ring: Middleware

- https://github.com/ring-clojure/ring-defaults
- https://github.com/ring-clojure/ring-json

Ring

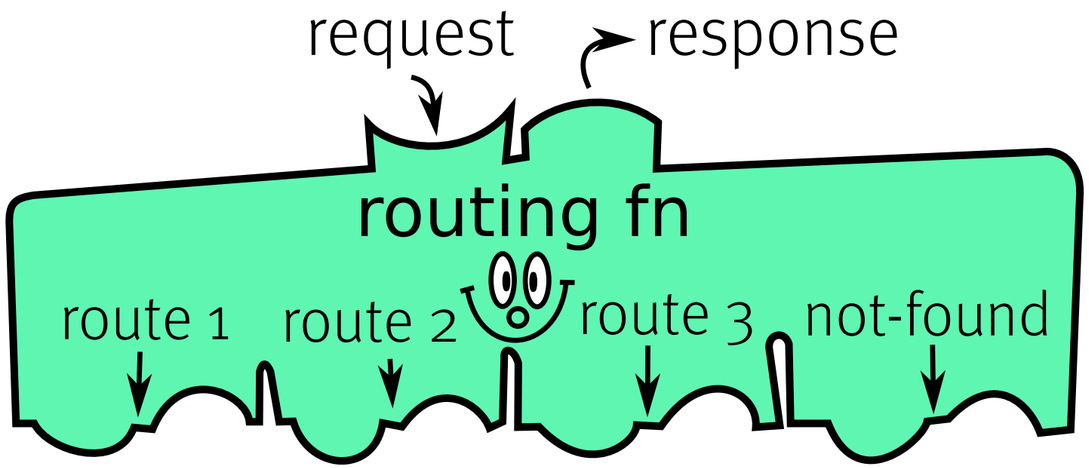

Java HTTP Server
Ring Server Adapter
middleware
middleware
webapp

DEVOXX™

Compojure

https://github.com/weavejester/compojure

Compojure


request
response
routing fn
route 1
route 2
route 3
not-found

Compojure: Handler

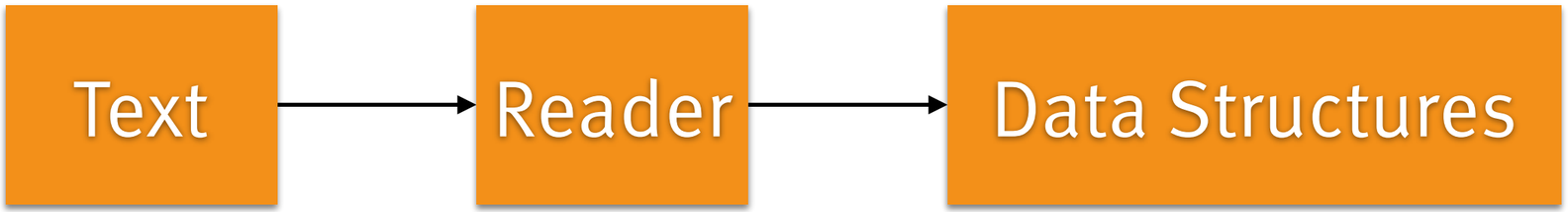
```
(def handler
  (GET “/hello” [] “Hello, world!”))

(handler {:request-method :get, :uri “/hello”})
› {:body “Hello, world!”}

(handler {:request-method :post, :uri “/hello”})
› nil
```

Excursion: Macros

Compojure: Macro




```
"(case 3
  1 "one"
  2 "two"
  "more")"
```

```
(case 3
  1 "one"
  2 "two"
  "more")
```

Compojure: Macro


```
(case 3
  1 “one”
  2 “two”
  “more”)
```

```
(if (= 3 1)
  “one”
  (if (= 3 2)
    “two”
    “more”))
```

DEVOXX™

Back to Compojure…

Compojure: Macro

```
(GET "/hello" [] "Hello, world!")

(fn [req]
  (if (and (= (:uri req) "/hello")
           (= (:request-method req) :get))
    {:body "Hello, world!"}))
```

Compojure: Path

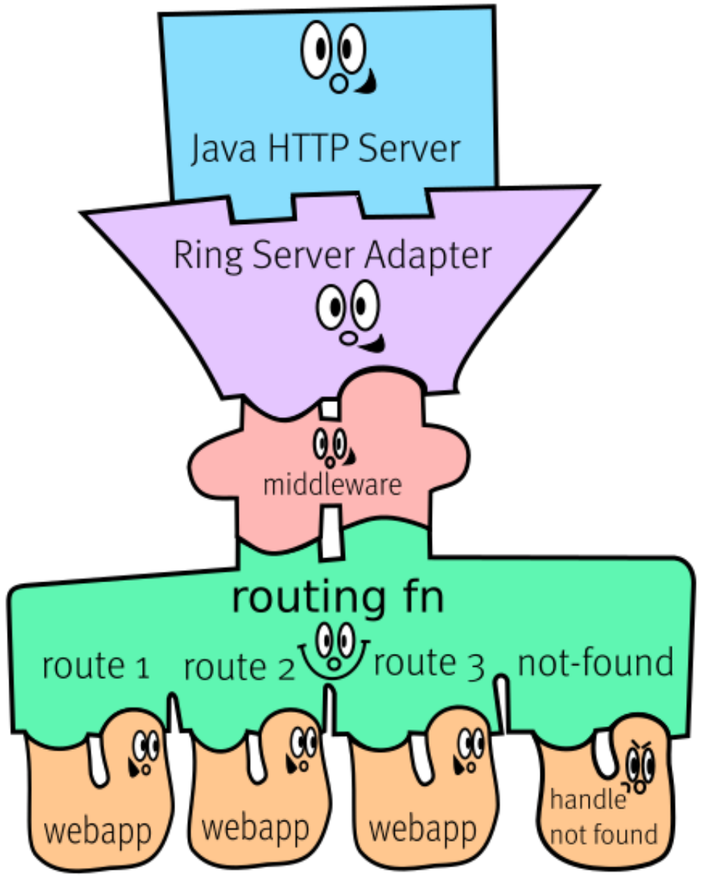
```
(GET "/hello/:name" [name]
  (str "Hello, " name "!"))
```

Compojure: Routes

```
(def my-routes
  (routes
    (GET “/hello” [] “Hello!”)
    (GET “/bye” [] “Bye!”)))

(defroutes my-routes
    (GET “/hello” [] “Hello!”)
    (GET “/bye” [] “Bye!”))
```

Compojure


Java HTTP Server
Ring Server Adapter
middleware
routing fn
route 1
route 2
route 3
not-found
webapp
webapp
webapp
handle
not found

Compojure: Alternatives

- https://github.com/juxt/bidi
- http://clojure-liberator.github.io/liberator/

DEVOXX™

Hiccup

https://github.com/weavejester/hiccup

Hiccup: Basics

```
[:div {:id “foo”}
  [:span “bar”]]
```

```
‹div id=“foo”›
  ‹span›bar‹/span›
‹/div›
```

Hiccup: Shortcuts

```
[:div#foo
  [:span “bar”]]
```

```
‹div id=“foo”›
  ‹span›bar‹/span›
‹/div›
```

Hiccup: Links

```
(link-to “www.innoq.com” “innoQ”)

[:a {:href “www.innoq.com”} “innoQ”]

‹a href=“www.innoq.com”›
  innoQ
‹/a›
```

Hiccup: Forms

```
(form-to [:post "/login"]
  (text-field "Username")
  (password-field "Password")
  (submit-button "Login"))
```

Hiccup: Alternatives

- https://github.com/danlarkin/clabango
- https://github.com/plakat/clj-thymeleaf
- https://github.com/cgrand/enlive
- https://github.com/yogthos/Selmer

Wrap-Up

Wrap-Up

- Functional language fits to HTTP cycle
- Clojure usage increases
- Libraries vs. Frameworks
- Healthy and welcome community

Interesting libraries

- https://github.com/technomancy/leiningen
- https://github.com/weavejester/environ
- https://github.com/clojurewerkz/route-one
- https://github.com/krisajenkins/yesql
- https://github.com/seancorfield/clj-time

“Frameworks”

- https://github.com/weavejester/duct
- http://www.luminusweb.net
- https://github.com/otto-de/tesla-microservice
- https://github.com/metosin/compojure-api