

Clojure & Web

Introduction to functional web

Clojure & Web

Tools & Setup

Demo 1: Jetty

Basic Clojure Web - Ring, Hiccup

Request Routing - Compojure

Demo 2: Ring, Hiccup, Compojure

Demo 3: Simple chat application

ClojureScript

Demo 4: Chat with Xhrlo

Heroku

Resources

This presentation <http://goo.gl/nEFWB>

Fork [lambda-prague/clojure-web-demo](#) on GitHub

Leiningen



Leiningen is a tool which simplifies setup and tasks inside Clojure projects (something like make and apt-get if you are familiar with Linux command line)

Setup

Download leiningen [script](#) from and put on your PATH

project.clj

Project configuration and dependencies described in clojure data-structures. See examples in demo.

Commands for this session

```
lein new <project name>
```

```
lein deps
```

```
lein repl
```

Additional resources

<http://leiningen.org/>, [Tutorial on GitHub](#)

Tools & Setup

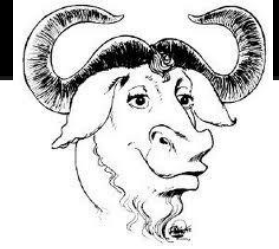
Ring & Hiccup

Request routing

ClojureScript

Heroku deploy

Clojure & Emacs



Clojure mode

Syntax highlighting, indentation, interaction with Slime.

Inferior lisp

Standard Emacs mode for interacting with Lisp-like REPL tools (part of Emacs)

```
C-u M-x inferior-lisp RET lein repl RET
```

Slime & Swank

Slime is advanced Emacs package for interacting with Lisp REPL. Clojure backend is distributed as a part of lein-swank package. Slime is now officially obsolete.

```
project.clj:    {:plugins [[lein-swank "1.4.4"]]}
Emacs          M-x clojure-jack-in RET
```

nRepl - Clojure network REPL

tools.nrepl is new standard for interacting with running Clojure programs in REPL way. There are different backends: nrepl.el, Counterclockwise (Eclipse) etc.

```
Emacs          M-x nrepl-jack-in
```

Clojure & Emacs



Alternatives

[Ritz](#) for advanced debugging.

Useful Emacs modes

[paredit-mode](#) for editing always balanced parens

[highlight-parentheses](#) - rainbow like highlighting

[yasnipet](#) - powerful template system for Emacs

Installation

Using [ELPA](#) - part of Emacs 24

Tools & Setup

Ring & Hiccup

Request routing

ClojureScript

Heroku deploy

Jetty & Ring [see demo-1]



Jetty

Open source HTTP server implemented in Java

Ring

Web applications library inspired by Python's WSGI and Ruby's Rack.

project.clj

```
[ring/ring-jetty-adapter "1.1.5"]  
[ring/ring-core           "1.1.5"]  
[ring/ring-devel          "1.1.5"]
```

src/demo/core.clj

```
(require '[ring.adapter.jetty :as jetty])  
  
(defn hello-world [REQ] {:status 200, :body "Hello, world!"})  
  
(jetty/run-jetty #'hello-world  
                  {:join?      false,  
                   :port      PORT})
```

Resources

[Jetty](#) [Ring on GitHub](#)

Tools & Setup

Ring & Hiccup

Request routing

ClojureScript

Heroku deploy

Hiccup [see demo-2]

Observation - HTML is just talkative incarnation of Lisp S-expressions.
Forget about writing HTML templates, let's write some Lisp.

Hiccup

Library for representing HTML in Clojure.

```
(html [:div#foo.bar.baz {:data-test "foobar"} "Some text"])  
  
=> <div class="bar baz" data-test="foobar" id="foo">Some text</div>
```

Demo 2

Now let's see what you can build with Hiccup :-)

Resources

[Hiccup on GitHub](#)

Tools & Setup

Ring & Hiccup

Request routing

ClojureScript

Heroku deploy

Compojure [see demo 2 & 3]

Compojure is DSL for routing library for Ring.

defroutes

Generates Ring handler functions

```
(defroutes main-route
  (GET "/" []
    {:status 200, :body "Hello, world!"})
```

destructuring

```
(GET "/room/:room" [{room :room} :params]
  ....)

(POST "/room/:room" [room :as req]
  ....)

(context "/room" []
  ....)
```

Resources

[Compojure on GitHub](#), [Liberator](#) - Clojure library for building RESTful applications

Tools & Setup

Ring & Hiccup

Request routing

ClojureScript

Heroku deploy

Ring middleware [see demo 3]

Idea: jetty handler is a function with one parameter (a request describing map).
We can build a set of functions which affect request resolution and combine them together

```
(defn wrap-function [FN]
  (fn [REQ]
    (FN (my-funky-req-function REQ))))
```

ring.middleware

- Content type handling
- Handling parameters
- Session management
- Development (stack traces)
- etc, etc, etc

Resources

[Ring Wiki](#), [Ring documentation](#)

Tools & Setup

Ring & Hiccup

Request routing

ClojureScript

Heroku deploy

ClojureScript [see main demo]

Benefits

- one language for server & browser
- access to JavaScript
- built using Google Closure - advanced compilation
- standard Clojure data structures(records, PersistentVector, PersistentHashMap...)
- REPL from editor
- can be used for server as well or as a query language for various NoSQL DB

Not 100% Clojure

- No support for parallel execution (no STM, promise, future)
- Differences between Java and JavaScript: (`< 0 nil`) \Rightarrow true

lein-cljsbuild

- Leiningen plugin which helps ClojureScript integration
- supports shared code (crossovers)

Resources

[Google Closure](#), [ClojureScript on GitHub](#), [lein-cljsbuild on GitHub](#)

Tools & Setup

Ring & Hiccup

Request routing

ClojureScript

Heroku deploy

Clojure hosting in clouds

Google App Engine

- hosting on Google infrastructure as Java project.
- appengine-magic helps different tasks

Heroku

- support for multiple languages (Ruby, Node.js, Clojure, Java, Python, Scala)
- pluggable - there is even configuration for Emacs lisp
- you deploy directly .git repository (no jar files)

Resources

Clojure [appengine-magic on GitHub](#), [Clojure on Heroku Dev Center](#)

Tools & Setup

Ring & Hiccup

Request routing

ClojureScript

Heroku deploy

Heroku deployment

Prerequisites

Heroku account, heroku utils and upload ssh key.

Procfile

```
web: lein trampoline run -m demo.core $PORT
```

project.clj

Setup leiningen hooks for ClojureScript

```
:hooks [leiningen.cljsbuild]
```

setup

```
$ heroku app:create my-app
```

```
$ heroku labs:enable user_env_compile -a my-app
```

```
$ heroku config:add --app my-app LEIN_BUILD_TASK="compile :all"
```

```
$ git remote add heroku git@heroku.com:my-app.git
```

```
$ git push heroku master
```

Tools & Setup

Ring & Hiccup

Request routing

ClojureScript

Heroku deploy

Use the source, Luke :)

(begin)

This could be the start of very beautiful program. (Slime)