



nklein

Weblocks

An Advanced Web Framework in Common Lisp

Patrick Stein / TC Lisp Users Group / 2010-04-26

- Originally written by Slava Akhmechet
 - <http://www.defmacro.org/>
- Now developed and maintained by Stephen Compall and Leslie Polzer
- “Whenever I have to write convoluted code to express a simple idea, I instinctively try to avoid doing work. The corollary of this is that every time I reach for my inbox during coding, there’s a pretty good chance I need an abstraction I don’t have.”

Embraces Lisp

- Multiple Dispatch
- Metaobject Protocol
- Lexical Closures
- Keyword Arguments
- Macros
- Continuations

Feature: Abstractions

```
(defun init-user-session (c)
  (setf (widget-children c)
        (make-top-page)))

(defun make-top-page ()
  (make-navigation
   'my-navigation-widget
   'main (make-blog-widget)
   'admin (make-admin-page)))
```

Think and code in
high level
abstractions

pages & widgets

pages are a
composite of
widgets

widgets maintain
their own state

Feature: Newbie Friendly

Easy to use and
install... if you use
clbuild

Took me hours
to get going with
Lispy and
ASDF-Install

```
closer-mop, metatilities,  
hunchentoot, cl-who,  
cl-ppcre, cl-json, puri,  
md5, cl-fad, fare-matcher,  
cl-cont, parenscrip,  
anaphora, f-underscore,  
bordeaux-threads,  
salza2, trivial-timeout
```

```
moptilities, cl-containers,  
metabang-bind, cl-unicode,  
asdf-system-connections,  
fare-utils, rfc2388, usocket,  
flexi-streams, chung,  
cl-base64, cl+ssl, (more...)
```

Feature: Multiple Views

```
(defclass message ()
  (id :initarg :id)
  (timestamp :reader ...)
  (body :initarg :body
        :accessor ... :type ...))

(defview msg-tview
  (:type table
   :inherit-from
    '(:scaffold message))
  (id :hidep t))

(defview msg-summary-tview
  (:type table
   :inherit-from msg-tview)
  (body :reader #'msg-summary))
```

Define classes

Introspect to
create a decent
view of the data

Inherit from
another view and
override parts

Feature: Continuations

Continuations let
you make flow
wizards easily

Use
(YIELD ...) to show each
page

Use
(ANSWER ...) to advance

```
(defun init-user-session (c)
  (with-flow c
    (if (and (yield #'ag-1)
              (yield #'ag-2)
              (yield #'ag-3))
        (show-protected-info)
        (show-error))))
```

```
(defun ag-1 (k)
  (with-html
    (:p "Agree to this!")
    (render-link
      #'(lambda (&rest args)
          (answer k t))
      "I Agree")))
```

Feature: Easy Ajax

- AJAX enabled by default
 - Can explicitly turn it off for individual widgets
- Seamlessly falls back on full-page reloads if JavaScript is disabled
- Widgets use MOP to know when they are dirty so they know when they need to update the display

Feature: Dispatching

```
".../foo/bar/baz.html" =>  
'( "foo" "bar" "baz.html" )  
  
(defwidget my-widget  
  (on-demand-selector)  
  (:default-initargs  
   :lookup-function  
   #'my-picker))  
  
(defun my-picker (self toks)  
  ;; any processing of the  
  ;; tokens you desire  
  (values widget-to-show  
          tokens-consumed  
          tokens-leftover))
```

URL gets broken
into tokens

Picker function
can consume any
of those tokens
and return a
widget to display

Can return a
widget with its
own picker

Feature: It's Lisp

- Fully extensible with CLOS
 - :before, :around, :after
 - extending classes
- Macros
 - DRY (Don't Repeat Yourself)

;;; Side-effect of DRY: you sometimes have
;;; to play with Hunchentoot directly

;;; like when I didn't want to return HTML:

```
(setf (hunchentoot:content-type*) "text/plain")  
(hunchentoot:abort-request-handler  
  "return this instead")
```

- <http://www.lamsight.org/>
 - Coordinating LAM research (LAM is a rare disease in reproductive-age females that causes muscle-like growth in the lungs)
- <http://www.thanadar.de/>
 - Browser-based RPG (in German)
- <http://aulapolska.pl/>
 - Appears to be some Polish gov't org that manages grants from EU

- I uncovered a bug on a Wednesday
- It was fixed in source control by Thursday
- Several other such reports happened this month

Can make
widgets out of
just strings or
functions or from
classes

Views generated
via introspection

Use MOP to
know when dirty

```
(make-widget "Some text")  
(make-widget  
  #'(lambda (&rest a)  
        (declare (ignore a))  
        (with-html  
          (:h2 "Title")  
          (:p "Body text")))))  
  
(defview my-form  
  (:type form  
   :inherit-from  
   '(:scaffold my-class)))
```


Quickstart Template

```
(wop:make-app  
  'my-app-name  
  "directory/for/app/")
```

Quick shell of a
web app with one
command

- Documentation/Examples
 - If you want editable/sortable table views or multiple such views with a navigation menu, you're golden
 - Beyond that, you've got to scour the mailing list to see who did it last
- (make-widget function-designator) captures the function rather than the symbol
- Had to restart user sessions often to test new functions.

- **Mailing List**

- <http://groups.google.com/group/weblocks>

- **Various intro docs**

- <http://defmacro.org/ramblings/ui-dsl.html>
- <http://defmacro.org/ramblings/continuations-web.html>

- **Various tutorials & TINAA docs**

- <http://weblocks.viridian-project.de/documentation>