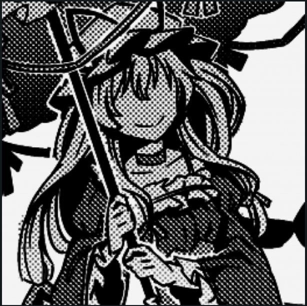

Nicolas Hafner

@Shinmera

https://everything.shinmera.com

Radiance - A Shiny Web Framework for Common Lisp

Usual Framework Focus

- Single application
- Tightly integrated
- Convenience through large toolsets

Usual Framework Focus

- Single application
- Tightly integrated
- Convenience through large toolsets
- Often force libraries on you

Usual Framework Focus

- Single application
- Tightly integrated
- Convenience through large toolsets
- Often force libraries on you
- Problematic to host multiple applications

Radiance’s Goals

- Many applications simultaneously
- Shared resources
- Exchangeable parts
- Easy to use and deploy

Radiance’s Goals

- Many applications simultaneously
- Shared resources
- Exchangeable parts
- Easy to use and deploy
- Fully covered by a specification

```
(define-interface cache
  (defun get (name))
  (defun renew (name))
  (defmacro with-cache
      (name-form test-form &body forms)))
```

```
;; ...
(defun cache:get (name)
  (read-data-file (cache::file name)))

(defun cache:renew (name)
  (delete-file (cache::file name)))

(defmacro cache:with-cache
    (name test &body forms)
  (let ((cache (gensym "CACHE")))
    `(let ((,cache ,name))
       (if (or (not (cache::exists ,cache))
               ,test)
           (cache::output ,cache
             ((lambda () ,@forms)))
           (cache:get ,cache)))))
```

```
(db:create 'vote '((option (:varchar 32))
                   (votes :integer)))

(define-page display #@"/" ()
  (r-clip:process T (dm:get 'vote)))
```

radiance-logo.png

https://github.com/Shirakumo/radiance

```
(ql-dist:install-dist
 "http://dist.tymoon.eu/shirakumo.txt")
(ql:quickload :radiance)
(radiance:startup)
```