



Nicolas Hafner
@Shinmera

<https://everything.shinmera.com>

Radiance - A Web Framework Environment 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
- Only use what you want

Radiance's Goals

- Many applications simultaneously
- Shared resources
- Exchangeable parts
- Only use what you want
- Fully specified

A Tiny Example

- option: Pizza votes: 32
- option: Pasta votes: 4
- option: Lasagna votes: 12
- option: Crêpes votes: 9
- option: Mashed Potatoes votes: 2

Submit


```
(in-package #:rad-user)
(db:create 'vote '((option (:varchar 32))
                   (votes :integer)))

(define-page display #@"/" ()
  (r-forms:choose
    #@"/api/vote" "id"
    (dm:get 'vote (db:query :all))))

(define-api vote (id) ()
  (with-model-save model ('vote (:= '_id id))
    (votes)
    (incf votes))
  (redirect #@"/")
  (api-output "Vote registered."))
```

```
(define-interface (database db)
  (defun create (collection structure
                 &key indices if-exists))
  (defmacro query (query-form))
  ...))

(define-interface (data-model dm)
  (defun get (collection query
             &key (skip 0) (amount 0) sort))
  ...))
```

```
(define-module #:my-database
  (:use #:cl)
  (:implements #:database))
(in-package #:my-database)
```

```
(defun db:create (database db
                  &key indices if-exists)
  ...)
```

```
(defmacro db:query (query-form)
  ...)
```

```
...
```

What Radiance Offers

Modules

Interfaces

Request handling

URIs

URI Patterns

Resources

Bidirectional Routing

Dispatching

What Radiance Offers

Modules

Interfaces

Request handling

URIs

URI Patterns

Resources

Bidirectional Routing

Dispatching

Banning

Rate limiting

Administration interface

Cache handling

Authentication

Sessions

User storage

User profiles

Logging

Database interaction

DB abstractions

Templating

Error handling

What Radiance Offers

Modules

Interfaces

Request handling

URIs

URI Patterns

Resources

Bidirectional Routing

Dispatching

Banning

Rate limiting

Administration interface

Cache handling

Authentication

Sessions

User storage

User profiles

Logging

Database interaction

DB abstractions

Templating

Error handling

Chatlog

Blog software

File hosting

Imageboard

Database introspection

Radiance

<https://github.com/Shirakumo/radiance>

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