



REDIS

Bruno Michel

<bruno.michel@af83.com>

42 bd de Sébastopol
75003 Paris

1436 "A" Howard Street,
San Francisco, California 94103

contact@af83.com
www.af83.com

🌀 REDIS = REmote DIctionary Server

🌀 Une base de données de type clé=valeur

- Clé = une chaîne de caractère
- Valeur de plusieurs types
 - Chaînes de caractères
 - Listes
 - Ensembles (Set)
 - None

🌀 Concurrents

- Memcachedb
- Tokyo Cabinet

Très rapide

- Écrit en C
- Écritures asynchrones sur le disque

Opérations de haut niveau

- Ajouter/Supprimer des éléments dans une liste
- Faire l'intersection de plusieurs ensembles
- Incrémenter un compteur et récupérer la valeur

Simple à comprendre

Hype & Fun ;-)

Encore très jeune

- Projet commencé fin février 2009
- Version 7 bêta

Ne remplace pas des requêtes SQL

Il faut apprendre certaines techniques

 REDIS

- `svn co http://redis.googlecode.com/svn/trunk redis`
- `cd redis && make`
- `./redis-server`

 Redis-rb (Ruby)

- `git clone git://github.com/ezmobius/redis-rb.git`
- `cd redis-rb && rake package`
- `gem install pkg/redis*.gem`

 Existe aussi pour PHP, Python et Erlang

- `http://code.google.com/p/redis/source/browse/trunk/client-libraries/`

Se connecter

- `require 'rubygems'`
- `require 'redis'`
- `redis = Redis.new`

Un couple clé/valeur

- `redis['foo'] = 'bar'`
- `redis['foo'] # => 'bar'`
- `redis.key?('foo') # => true`
- `redis.delete('foo')`

Incrémenter et Décrémenter

- `5.times { p redis.incr('counter') }`
- `5.times { p redis.decr('counter') }`

Listes

- `redis.push_head('ma_liste', 'foo')`
- `redis.push_tail('ma_liste', 'bar')`
- `redis.list_range('ma_liste', 0, -1) # => ['foo', 'bar']`
- `redis.list_length('ma_liste') # => 2`
- `redis.pop_tail # => 'bar'`
- `redis.pop_head # => 'foo'`

Ensembles

- `redis.set_add('my_tags', 'foo')`
- `redis.set_add('my_tags', 'bar')`
- `redis.set_members('my_tags') # => Set['foo', 'bar']`
- `redis.set_count # => 2`
- `redis.set_intersect('my_friends', 'your_friends')`

Namespaces

- `redis['users:1000:login'] = 'joe'`
- `redis['users:1000:email'] = 'joe@example.com'`
- `redis['posts/1234/title'] = '/'` peut aussi faire séparateur
- `redis['comments_1234567'] = "Tout sauf ' ' et '\n'"`
- `redis.keys('users*:login')`

ID auto-incrémenté

- `uid = redis.incr('users:nextId')`
- `redis["users:#{uid}:login"] = 'joe'`
- `redis.push_tail('users:all', uid)`

Trier une liste

- `redis.sort('ma_liste')`
- `redis.sort('users',
:by => 'users*:nb_followers',
:order => 'DESC',
:limit => [0, 10])`

Buffer circulaire

- `redis.push_tail('logs', 'some log line..')`
- `redis.list_trim('logs', 0, 99)`

Objectif

- Faire une application web
- Avec Redis (bien entendu)
- Dans le langage/framework de votre choix
 - PHP
 - Python
 - Ruby on Rails
 - Sinatra
- Par groupe de deux

DevLinks

- Poster des liens : URL +description
- Tagger les liens

DevIdées

- Proposer des idées
- Voter +1 ou -1
- Trier les idées par popularité

Notifications

- Poster des messages de notification
- Afficher une fois les notifications

af83

af83

Licence : CC by-sa 2.0+

<http://creativecommons.org/licenses/by-sa/2.0/fr/>

Paris Office

42, bd Sébastopol
73 003 Paris
France
+33 1 40 27 83 83

San Francisco Office

1436 "A" Howard Street,
San Francisco, Californi 94103
United States
+1 415 578 4664

www.af83.com
contact@f83.com