

Javascript, The Swiss Army Knife of Programming Languages

David Morcillo

30-11-2013

Persistence

We stored player positions on a simple object on the server-side code. If we reboot the server we will lose these values.

Simple object lost when server restarts

```
var playerPositions = {  
  player1: { x: 100, y: 200 },  
  player2: { x: 100, y: 400 },  
  player3: { x: 700, y: 200 },  
  player4: { x: 700, y: 400 }  
};
```

RedisDB

Website definition

Redis is an open source, BSD licensed, advanced key-value store. It is often referred to as a data structure server since keys can contain strings, hashes, lists, sets and sorted sets.



RedisDB: Installation

From source code

```
$ wget http://download.redis.io/redis-stable.tar.gz
$ tar xvzf redis-stable.tar.gz
$ cd redis-stable
$ make
```

RedisDB: Installation

From source code

```
$ wget http://download.redis.io/redis-stable.tar.gz
$ tar xvzf redis-stable.tar.gz
$ cd redis-stable
$ make
```

Copy executable and default configuration

```
$ sudo cp src/redis-server /usr/local/bin/
$ sudo cp src/redis-cli /usr/local/bin/
$ sudo cp redis.conf /etc/redis.conf
```

RedisDB: Installation

From source code

```
$ wget http://download.redis.io/redis-stable.tar.gz
$ tar xvzf redis-stable.tar.gz
$ cd redis-stable
$ make
```

Copy executable and default configuration

```
$ sudo cp src/redis-server /usr/local/bin/
$ sudo cp src/redis-cli /usr/local/bin/
$ sudo cp redis.conf /etc/redis.conf
```

Run server

```
$ redis-server
[28550] 01 Aug 19:29:28 # Warning: no config file specified, using the default config...
[28550] 01 Aug 19:29:28 * Server started, Redis version 2.2.12
[28550] 01 Aug 19:29:28 * The server is now ready to accept connections on port 6379
...
```

RedisDB: Client

Installation

```
npm install redis --save-dev
```


RedisDB: Client

Installation

```
npm install redis --save-dev
```

Create client on our code

```
var redis = require('redis'),  
    redisClient = redis.createClient();
```

RedisDB: Commands

In this workshop we are going to use one Redis data structure: a Hash

RedisDB: Commands

In this workshop we are going to use one Redis data structure: a Hash

Get all fields and values from a hash

```
redisClient.hgetall('myHash', function (err, result) {  
  // result is converted on a Javascript object  
});
```

Store some fields and values to a hash

```
redisClient.hmset(['myHash', 'key1', 'value1', 'key2', 'value2'], function (err, result) {  
  // code omitted  
});
```

RedisDB: Lab

- `git checkout stage_9`
- Install your back-end dependencies with `npm install`
- Install your front-end dependencies with `bower install`
- Start `grunt watch` for auto linting
- Find TODOs and complete the exercise.