Eredis

Using Redis from Erlang

What is Redis?

- Open source (BSD licence)
- NoSQL database (key-value storage)
- Stores data in RAM → it's fast!
- Used mainly for caching
- All operations on datatypes are atomic

written in C

Where to use it

- Cache
- Storing data that can be lost
- Expiring data persistance

Where rather not to use it?

- Big data storage
- Crucial information persistance

Eredis

- Non-blocking Redis client
- Open-source: `github.com/wooga/eredis`
- Focused on performance and robustness
- Supporting:
  - Any command to redis server
  - Transactions
  - Pipelining
  - Pubsub

Usage

- with rebar

```
┌─misqos@Acer-ubuntu  ~/workspace/erlang/eredis_intro <ruby-2.1.3> <master*>
└─$ cat rebar.config
{deps,
  [
    {eredis, ".*", {git, "https://github.com/wooga/eredis.git", "HEAD"}}
  ]
}.
┌─misqos@Acer-ubuntu  ~/workspace/erlang/eredis_intro <ruby-2.1.3> <master*>
└─$ rebar get-deps
==> eredis (get-deps)
==> eredis_intro (get-deps)
```

Usage

- without rebar

```
misqos@Acer-ubuntu  ~/workspace/erlang/eredis <ruby-2.1.3> <master*>
$ git clone git://github.com/wooga/eredis.git .
  /* output omited */
misqos@Acer-ubuntu  ~/workspace/erlang/eredis <ruby-2.1.3> <master*>
$ ./rebar compile
  /* output omited */
```

Running

```
┌─misqos@Acer-ubuntu  ~/workspace/erlang/eredis <ruby-2.1.3> <master*>
└─$ cat start.sh
erl -pa ebin -pa deps/*/ebin
┌─misqos@Acer-ubuntu  ~/workspace/erlang/eredis <ruby-2.1.3> <master*>
└─$ rebar compile
    /* output omited */
┌─misqos@Acer-ubuntu  ~/workspace/erlang/eredis <ruby-2.1.3> <master*>
└─$ ./start.sh
```

SET and GET

```
{ok, <<"OK">>} = eredis:q(C, ["SET", "foo", "bar"]).
{ok, <<"bar">>} = eredis:q(C, ["GET", "foo"]).
```

MSET, MGET

```
KeyValuePairs = ["key1", "value1", "key2", "value2", "key3", "value3"].
{ok, <<"OK">>} = eredis:q(C, ["MSET" | KeyValuePairs]).
{ok, Values} = eredis:q(C, ["MGET" | ["key1", "key2", "key3"]]).
```

Transactions

```
{ok, <<"OK">>} = eredis:q(C, ["MULTI"]).
{ok, <<"QUEUED">>} = eredis:q(C, ["SET", "foo", "bar"]).
{ok, <<"QUEUED">>} = eredis:q(C, ["SET", "bar", "baz"]).
{ok, [<<"OK">>, <<"OK">>]} = eredis:q(C, ["EXEC"]).
```

Examples

- GET/SET example `example.erl`
- Simple logging system `logger.erl`
- PubSub `subscribe.erl`