Redis - Wprowadzenie

Mirosław Boruta

Silesian Ruby User Group

19 lutego 2011

Redis jest...

REmote Dictionary Service

Redis jest...

Redis jest bazą przechowującą pary klucz-watrość, przy czym na wartości można dodatkowo wykonywać operacje.



Mocne strony Redis-a

Wydajność

Mocne strony Redis-a

- Wydajność
- Atomiczność podstawowych operacji

Mocne strony Redis-a

- Wydajność
- Atomiczność podstawowych operacji
- Wbudowane typy danych



Słabe strony Redis-a

Trwałość danych

Słabe strony Redis-a

- Trwałość danych
 - RAM

Słabe strony Redis-a

- Trwałość danych
- RAM
- Optymalizacja pod względem szybkości kosztem zajmowanej przestrzeni pamięci

Klucze są ciągami znaków:

Klucze są ciągami znaków:

redis> SET foo bar OK

Klucze są ciągami znaków:

```
redis> SET foo bar
OK
redis> GET foo
"bar"
```

Klucze są ciągami znaków:

```
redis> SET foo bar
OK
redis> GET foo
"bar"
redis> GET baz
(nil)
```

Klucze mogą zawierać nie tylko litery:

Klucze mogą zawierać nie tylko litery:

redis> SET user:1:name Bob OK

Klucze mogą zawierać nie tylko litery:

```
redis> SET user:1:name Bob
OK
redis> SET user:1:e-mail bob@ex
```

Pobieranie listy kluczy na podstawie wzorca:

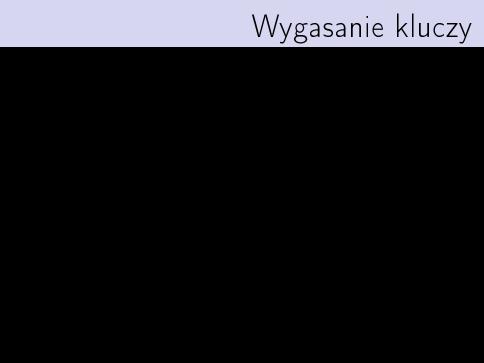
Pobieranie listy kluczy na podstawie wzorca:

```
redis> KEYS user*
```

- 1. "user:1:name"
- 2. "user:1:e-mail"

Pobieranie listy kluczy na podstawie wzorca:

```
redis> KEYS user*
1. "user:1:name"
2. "user:1:e-mail"
redis> KEYS user:*:name
1. "user:1:name"
```



Wygasanie kluczy

redis> EXPIRE foo 2 (integer) 1

Wygasanie kluczy

```
redis> EXPIRE foo 2
(integer) 1
```

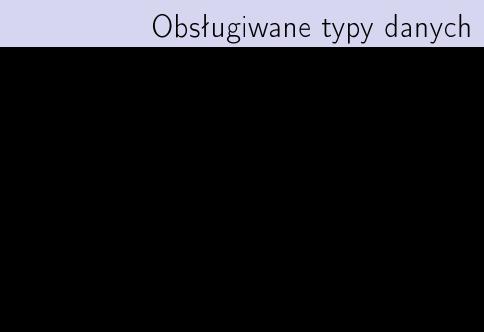
2 sekundy później...

Wygasanie kluczy

```
redis> EXPIRE foo 2
(integer) 1
```

2 sekundy później...

```
redis> GET foo
(nil)
```



Ciąg znaków (String)

- Ciąg znaków (String)
- Lista (*List*)

- Ciąg znaków (String)
- Lista (List)
- Zbiór (Set)

- Ciąg znaków (String)
- Lista (List)
- Zbiór (Set)
- Zbiór sortowany (Sorted set)

- Ciąg znaków (String)
- Lista (List)
- Zbiór (Set)
- Zbiór sortowany (Sorted set)
- Hash (Hash)

Przydatne operacje na ciągach znaków:

Przydatne operacje na ciągach znaków:

SET, GET

Przydatne operacje na ciągach znaków:

SET, GET, MSET, MGET

Przydatne operacje na ciągach znaków:

- SET, GET, MSET, MGET
- INCR, DECR

Przydatne operacje na ciągach znaków:

- SET, GET, MSET, MGET
- INCR, DECR, INCRBY, DECRBY

Ciąg znaków

Przydatne operacje na ciągach znaków:

- SET, GET, MSET, MGET
- INCR, DECR, INCRBY, DECRBY
- SETEX

Ciąg znaków

Przydatne operacje na ciągach znaków:

- SET, GET, MSET, MGET
- INCR, DECR, INCRBY, DECRBY
- SETEX
- SETNX

redis> SET foo 0

OK

```
redis> SET foo 0
OK
redis> INCR foo
(integer) 1
```

```
redis> SET foo 0
OK
redis> INCR foo
(integer) 1
redis> INCRBY foo 2
(integer) 3
```

```
redis> SET foo 0
OK
redis> INCR foo
(integer) 1
redis> INCRBY foo 2
(integer) 3
redis> GET foo
11311
```

Ciąg znaków – przykład

Ciąg znaków – przykład

redis> SETEX 60 token h4x0rz

Przydatne operacje na listach:

LPUSH, RPUSH

- LPUSH, RPUSH
- LPOP, RPOP

- LPUSH, RPUSH
- LPOP, RPOP
- LRANGE

- LPUSH, RPUSH
- LPOP, RPOP
- LRANGE
- LTRIM

- LPUSH, RPUSH
- LPOP, RPOP
- LRANGE
- LTRIM
- RPOPLPUSH



redis> LPUSH newset a

redis> LPUSH newset a redis> LRANGE newset 0 -1

_____ 1. "a"

redis> LPUSH newset a redis> LRANGE newset 0 -1 1. "a" redis> LPUSH newset b

redis> LPUSH newset a
redis> LRANGE newset 0 -1
1. "a"
redis> LPUSH newset b
redis> RPUSH newset c

```
redis> LPUSH newset a
redis> LRANGE newset 0 -1
1. "a"
redis> LPUSH newset b
redis> RPUSH newset c
redis> LRANGE newset 0 -1
1. "b"
2. "a"
3. "c"
```

redis> RPOPLPUSH newset newset "c"

```
redis> RPOPLPUSH newset newset
"c"
redis> LRAGNE newset 0 -1
1. "c"
2. "b"
3. "a"
```

```
redis> RPOPLPUSH newset newset
^{II} ^{C} ^{II}
redis> LRAGNE newset 0 -1
1. "c"
2. "b"
3. "a"
redis> RPUSH newset d
```

```
redis> RPOPLPUSH newset newset
11 6 11
redis> LRAGNE newset 0 -1
1. "c"
2. "b"
3. "a"
redis> RPUSH newset d
redis> RPOPLPUSH newset newset
"d"
```

```
redis> LLEN newset (integer) 4
```

```
redis> LLEN newset
  (integer) 4
redis> LPUSH newset e
```

```
redis> LLEN newset
(integer) 4
redis> LPUSH newset e
redis> LTRIM newset 0 3
```

```
redis> LLEN newset
(integer) 4
redis> LPUSH newset e
redis> LTRIM newset 0 3
redis> LRANGE newset 0 -1
1. "e"
2. "d"
3. "c"
4. "b"
```

Przydatne operacje na zbiorach:

SADD, SREM

- SADD, SREM
- SMEMBERS, SRANDMEMBER, SPOP

- SADD, SREM
- SMEMBERS, SRANDMEMBER, SPOP
- SISMEMBER

- SADD, SREM
- SMEMBERS, SRANDMEMBER, SPOP
- SISMEMBER
- SDIFF, SDIFFSTORE

Zbiór

Przydatne operacje na zbiorach:

- SADD, SREM
- SMEMBERS, SRANDMEMBER, SPOP
- SISMEMBER
- SDIFF, SDIFFSTORE
- SINTER, SINTERSTORE

Zbiór

Przydatne operacje na zbiorach:

- SADD, SREM
- SMEMBERS, SRANDMEMBER, SPOP
- SISMEMBER
- SDIFF, SDIFFSTORE
- SINTER, SINTERSTORE
- SUNION, SUNIONSTORE



 $A = \{ 1, 2 \}, B = \{ 2, 3 \}$

```
A = { 1, 2 }, B = { 2, 3 } redis> SDIFF A B 1. "1"
```

```
A = { 1, 2 }, B = { 2, 3 }
redis> SDIFF A B
1. "1"
redis> SINTER A B
1. "2"
```

```
A = \{ 1, 2 \}, B = \{ 2, 3 \}
redis> SDIFF A B
1. "1"
redis> SINTER A B
1. "2"
redis> SUNION A B
1. "3"
2. "1"
```

3. "2"

Przydatne operacje:

ZADD, ZREM

- ZADD, ZREM
- ZINCRBY

- ZADD, ZREM
- ZINCRBY
- ZSCORE, ZRANK, ZREVRANK

- ZADD, ZREM
- ZINCRBY
- ZSCORE, ZRANK, ZREVRANK
- ZRANGE, ZREVRANGE

- ZADD, ZREM
- ZINCRBY
- ZSCORE, ZRANK, ZREVRANK
- ZRANGE, ZREVRANGE
- ZRANGEBYSCORE, ZREVRANGEBYSCORE

redis> ZADD days 1 Monday

redis> ZADD days 1 Monday redis> ZADD days 2 Tuesday

```
redis> ZADD days 1 Monday
redis> ZADD days 2 Tuesday
[...]
```

```
redis> ZADD days 1 Monday
redis> ZADD days 2 Tuesday
[...]
redis> ZRANGE days 0 -1
1. "Monday"
2. "Tuesday"
\lceil \ldots \rceil
7. "Sunday"
```

Przydatne operacje na hashach:

HSET, HMSET

- HSET, HMSET
- HGET, HMGET

- HSET, HMSET
- HGET, HMGET
- HDEL, HEXISTS

- HSET, HMSET
- HGET, HMGET
- HDEL, HEXISTS
- HGETALL, HKEYS, HVALS



redis> HSET user:1 name Bob

redis> HSET user:1 name Bob
redis> HSET user:1 email bob@ex

redis> HSET user:1 name Bob

"Bob"

redis> HGET user:1 name

redis> HSET user: 1 email bob@ex

redis> HSET user:1 name Bob redis> HSET user:1 email bob@ex redis> HGET user:1 name "Bob" redis> HGETALL user:1 1. "name" 2. "Bob"

4. "bob@example.com"

3. "email"

Transakcje w Redis pozwalają zgrupować wiele operacji i wykonać je jako jedną atomiczną operację

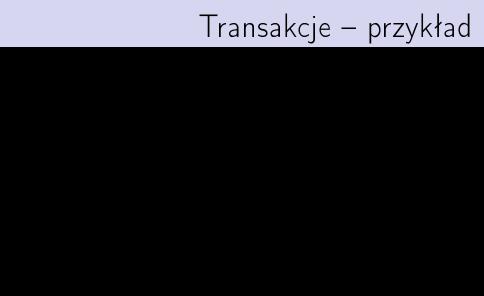
Polecenia operujące na transakcjach:

Polecenia operujące na transakcjach:

MULTI, EXEC, DISCARD

Polecenia operujące na transakcjach:

- MULTI, EXEC, DISCARD
- WATCH, UNWATCH



Transakcje – przykład

redis> MULTI

Transakcje – przykład

redis> MULTI redis> SET foo O QUEUED

Transakcje – przykład

```
redis> MULTI
redis> SET foo 0
QUEUED
redis> INCR foo
```

```
redis> MULTI
redis> SET foo O
QUEUED
redis> INCR foo
QUEUED
```

```
redis> MULTI
redis> SET foo 0
QUEUED
redis> INCR foo
QUEUED
redis> EXEC
1. OK
(integer) 1
```

Problem

redis> GET foo QUEUED

Rozwiązanie

Dostępne dopier od wersji 2.1.0

```
redis> WATCH foo
redis> GET foo
redis> MULTI
[...]
redis> EXEC
```

Przydatne polecenia:

Przydatne polecenia:

PUBLISH

Przydatne polecenia:

- PUBLISH
- SUBSCRIBE, UNSUBSCRIBE

Przydatne polecenia:

- PUBLISH
- SUBSCRIBE, UNSUBSCRIBE
- PSUBSCRIBE, PUNSUBSCRIBE

Pub/Sub – przykład producenta

Pub/Sub – przykład producenta

```
redis> PUBLISH msgs Hi!
(integer) 1
```

Pub/Sub – przykład producenta

```
redis> PUBLISH msgs Hi!
(integer) 1
redis> PUBLISH msgs Hi
(integer) 0
```

```
redis> SUBSCRIBE
```

- 1. "subscribe"
- 2. "msgs"
- 3. (integer) 1

```
redis> SUBSCRIBE
1. "subscribe"
2. "msgs"
3. (integer) 1
```

2. "msgs"

1. "message"

3. "Hi!"

```
redis> SUBSCRIBE
1. "subscribe"
2. "msgs"
3. (integer) 1
1. "message"
2. "msgs"
3. "Hi!"
redis> UNSUBSCRIBE
```

Podsumowanie możliwości Redis-a:

Szybka baza danych

- Szybka baza danych
- Przechowuje ciągi znaków, listy, zbiory, zbiory sortowane i hashe

- Szybka baza danych
- Przechowuje ciągi znaków, listy, zbiory, zbiory sortowane i hashe
- 💌 . . . i potrafi na nich wykonywać operacje

- Szybka baza danych
- Przechowuje ciągi znaków, listy, zbiory, zbiory sortowane i hashe
- 🔹 📖 i potrafi na nich wykonywać operacje
- Zawiera obsługę transakcji

- Szybka baza danych
- Przechowuje ciągi znaków, listy, zbiory, zbiory sortowane i hashe
- 🔹 📖 i potrafi na nich wykonywać operacje
- Zawiera obsługę transakcji
- Implementuje Pub/Sub

Koniec

To be continued...

Pytania?

Odnośniki

- http://redis.io
- http://blog.mjrusso.com/2010/10/17/ redis-from-the-ground-up.html
- Ruby gem: redis
- Node.js lib: gh:mranney/node_redis