Connect

CLI RedisInsight node-redis redis-py NRedisStack Jedis

```
# Syntax
redis-cli -u redis://host:port
redis-cli -u redis://username:password@host:port

# Examples
redis-cli
redis-cli -u redis://localhost:6379
redis-cli -u redis://myuser:mypassword@localhost:6379

# If you run Redis through Docker
docker exec -it <container-id-or-name> redis-cli
```

NOTE

To setup Redis either locally or in the cloud, refer to the tutorial

Strings/Numbers

CLI node-redis redis-py NRedisStack Jedis

Command	Syntax	Example	Output
SET	SET key value	`SET myKey "Hello"`	"OK"
Description: Set key to hold the string value. If key already holds a value, it is overwritten, regardless of its type.Time Complexity: O(1)			

GET	GET key	`GET myKey`	”Hello”
Description: Get the string value of key. If the key does not exist the special value nil is returned.Time Complexity: O(1)			
MGET	MGET key [key ...]	`MGET myKey nonExistentKey`	1) ”Hello” 2) (nil)
Description: Returns the values of all specified keys. For every key that does not hold a string value or does not exist, the special value nil is returned.Time Complexity: O(N)			
INCR	INCR key	`INCR myCounter`	(integer) 1
Description: Increments the number stored at key by one. If the key does not exist, it is set to 0 before performing the operation.Time Complexity: O(1)			

Generic

CLI node-redis redis-py NRedisStack Jedis

Command	**Syntax**	**Example**	**Output**
KEYS	KEYS pattern	`KEYS my*`	1) ”myKey” 2) ”myCounter”
Description: Returns all keys matching pattern.Time Complexity: O(N)			

EXISTS	EXISTS key [key ...]	`EXISTS myKey`	(integer) 1
Description: Checks if one or more keys exist.Time Complexity: O(N)			
EXPIRE	EXPIRE key seconds	`EXPIRE myKey 120`	(integer) 1
Description: Set a timeout on a key.After the timeout has expired, the key will automatically be deleted.Time Complexity:O(1)			
TTL	TTL key	`TTL myKey`	(integer) 113
Description: Returns the remaining time to live of a key that has a timeout.Time Complexity: O(1)			
PERSIST	PERSIST key	`PERSIST myKey`	(integer) 1
Description: Removes the expiration from a key.Time Complexity:O(1)			
SCAN	SCAN cursor [MATCH pattern] [COUNT count]	`SCAN 0 MATCH my* COUNT 2`	1) ”3” 2) 1) ”myCounter” 2) ”myKey”
Description: Iterates the set of keys in the currently selected Redis database.Time Complexity: O(1) for every call. O(N) for a complete iteration.			

DEL	DEL key [key ...]	DEL myKey	(integer) 1
Description: Removes the specified keys.Time Complexity: O(N)			
INFO	INFO [section]	INFO server INFO keyspace	# Server redis_version:6.2.5 redis_git_sha1:00000000 redis_build_id:9893b2a-dirty redis_mode:standalone os:Linux 5.4.72-microsoft-standard-WSL2 x86_64 arch_bits:64 ... # Keyspace db0:keys=2,expires=0,avg_ttl=0
server, clients, memory, persistence, stats, replication, cpu, commandstats, latencystats, sentinel, cluster, modules, keyspace, errorstats.Time Complexity: O(1)			

DEVELOPERS

→ Getting Started

→ Create >

↓ Develop ⌄

Commands cheat sheet

Overview

Node.js >

Python >

Hashes

CLI node-redis redis-py NRedisStack Jedis

Command	Syntax	Example	Output

HSET	HSET key field value [field value …]	`HSET h_employee_profile:101 name "Nicol" age 33`	(integer) 2
Description: Sets the specified fields to their respective values in the hash stored at key.Time Complexity: O(N)			
HGET	HGET key field	`HGET h_employee_profile:101 name`	”Nicol”
Description: Returns the value associated with field in the hash stored at key.Time Complexity: O(1)			
HGETALL	HGETALL key	`HGETALL h_employee_profile:101`	1) ”name” 2) ”Nicol” 3) ”age” 4) ”33”
Description: Returns all fields and values of the hash stored at key.Time Complexity: O(N)			
HMGET	HMGET key field1 [field2]	`HMGET h_employee_profile:101 name age`	1) ”Nicol” 2) ”33”
Description: Returns the values associated with the specified fields in the hash stored at key.Time Complexity: O(N)			

Sets

CLI node-redis redis-py NRedisStack Jedis

Command	Syntax	Example	Output
SADD	SADD key member [member ...]	`SADD mySet "Hello"`	(integer) 1
Description: Adds the specified members to the set stored at key. Time Complexity: O(N)			
SMEMBERS	SMEMBERS key	`SMEMBERS mySet`	1) "Hello"
Description: Returns all the members of the set value stored at key.Time Complexity: O(N)			
SCARD	SCARD key	`SCARD mySet`	(integer) 1
Description: Returns the set cardinality (number of elements) of the set stored at key.Time Complexity: O(1)			
SISMEMBER	SISMEMBER key member	`SISMEMBER mySet "Hello"`	(integer) 1
Description: Returns if member is a member of the set stored at key.Time Complexity: O(1)			
SDIFF	SDIFF key1 [key2]	`SDIFF mySet myOtherSet`	1) "Hello"

Description: Returns the members of the set resulting from the difference between the first set and all the successive sets.Time Complexity: O(N)			
SDIFFSTORE	SDIFFSTORE destination key1 [key2]	`SDIFFSTORE myNewSet mySet myOtherSet`	(integer) 1
Description: This command is equal to SDIFF, but instead of returning the resulting set, it is stored in destination.Time Complexity: O(N)			
SREM	SREM key member [member ...]	`SREM mySet "Hello"`	(integer) 1
Description: Removes the specified members from the set stored at key.			

Sorted sets

CLI node-redis redis-py NRedisStack Jedis

Command	Syntax	Example	Output
ZADD	ZADD key score member [score member ...]	`ZADD myZSet 1 "one" 2 "two"`	(integer) 2
Description: Adds all the specified members with the specified scores to the sorted set stored at key. Time Complexity: O(log(N))			
ZRANGE	ZRANGE key start stop [WITHSCORES]	`ZRANGE myZSet 0 -1`	1) ”one” 2)”two”

Description: Returns the specified range of elements in the sorted set stored at key.Time Complexity: O(log(N)+M) where M is the number of elements returned			

Lists

CLI node-redis redis-py NRedisStack Jedis

Command	Syntax	Example	Output
LPUSH	LPUSH key value [value ...]	`LPUSH myList "World"`	(integer) 1
Description: Inserts the specified values at the head of the list stored at key. Time Complexity: O(N)			
RPUSH	RPUSH key value [value ...]	`RPUSH myList "Hello"`	(integer) 2
Description: Inserts the specified values at the tail of the list stored at key.Time Complexity: O(N)			
LRANGE	LRANGE key start stop	`LRANGE myList 0 -1`	1) "World" 2) "Hello"
Description: Returns the specified elements of the list stored at key.Time Complexity: O(S+N) where S is the distance of start and N is the number of elements in the specified range.			
LLEN	LLEN key	`LLEN myList`	(integer) 2
Description: Returns the length of the list stored at key.Time Complexity: O(1)			

LPOP	LPOP key [count]	`LPOP myList`	”World”
Description: Removes and returns the first element of the list stored at key.Time Complexity: O(N)			
RPOP	RPOP key [count]	`RPOP myList`	”Hello”
Description: Removes and returns the last element of the list stored at key.Time Complexity: O(N)			

Streams

CLI node-redis redis-py NRedisStack Jedis

Command	Syntax	Example	
XADD	XADD key field value [field value ...]	`XADD myStream * sensorId "1234" temperature "19.8"`	15 0
Description: Appends the specified stream entry to the stream at the specified key. Time Complexi adding a new entry.			

XREAD	XREAD [COUNT count] [BLOCK milliseconds] STREAMS key [key ...] ID [ID ...]	`XREAD COUNT 2 STREAMS myStream 0`	1) 2) "1 0' "s "1 "t 4)
Description: Read data from one or multiple streams, only returning entries with an **ID greater** than received ID reported by the caller.			
XRANGE	XRANGE key start end [COUNT count]	`XRANGE myStream 1518951480106-0 1518951480106-0`	1) "1 0' "s "1 "t 4)
Description: Returns the entries matching a range of IDs in a stream. Time Complexity: O(N) with N b number of elements being returned. If N is constant (e.g. always asking for the first 10 elements wit can consider it O(1).			
XLEN	XLEN key	`XLEN myStream`	(ir
Description: Returns the number of entries of a stream. Time Complexity: O(1)			
XDEL	XDEL key ID [ID ...]	`XDEL myStream 1518951480106-0`	(ir
Description: Removes the specified entries from a stream. Time Complexity: O(1) for each single iten the stream			

XTRIM	XTRIM key MAXLEN [~] count	`XTRIM myStream MAXLEN 0`	(ir
Description: Trims the stream to a different length. Time Complexity: O(N), with N being the number entries. Constant times are very small however, since entries are organized in macro nodes containi entries that can be released with a single deallocation.			

ⓘ **REDIS STACK COMMANDS**

Redis stack extends the core features of Redis OSS like querying across hashes and JSON documents, time series data support, full-text search ..etc

JSON

CLI node-redis redis-py NRedisStack Jedis

Command	Syntax	Example
JSON.SET	JSON.SET key path value	`JSON.SET employee_profile:1 . '{"name":"Alice"`
Description: Sets JSON value at path in key.Time Complexity: O(M+N) where M is the original size anc		
JSON.GET	JSON.GET key [path [path ...]]	`JSON.GET employee_profile:1`
Description: Returns the JSON value at path in key.Time Complexity: O(N) when path is evaluated to when path is evaluated to multiple values, where N is the size of the key		

JSON.NUMINCRBY	JSON.NUMINCRBY key path number	JSON.SET employee_profile:1 .age 30 JSON.NUMINCRBY employee_profile:1 .age 5
Description: Increments a number inside a JSON document.Time Complexity: O(1) when path is evalu multiple values, where N is the size of the key		
JSON.OBJKEYS	JSON.OBJKEYS key [path]	JSON.OBJKEYS employee_profile:1
Description: Return the keys in the object that's referenced by path. Time Complexity: O(N) when p number of keys in the object, O(N) when path is evaluated to multiple values, where N is the size of		
JSON.OBJLEN	JSON.OBJLEN key [path]	JSON.OBJLEN employee_profile:1
Description: Report the number of keys in the JSON object at path in key. Time Complexity: O(1) whe is evaluated to multiple values, where N is the size of the key		
JSON.ARRAPPEND	JSON.ARRAPPEND key [path] value [value ...]	JSON.SET employee_profile:1 .colors '["red", " JSON.ARRAPPEND employee_profile:1 .colors '"ye
Description: Append the json values into the array at path after the last element in it. Time Complex added where N is the size of the key		
JSON.ARRINSERT	JSON.ARRINSERT key path index value [value ...]	JSON.ARRINSERT employee_profile:1 .colors 2 '"
Description: Insert the json values into the array at path before the index (shifts to the right). Time value where N is the size of the array, O(N) when path is evaluated to multiple values, where N is the		

JSON.ARRINDEX	JSON.ARRINDEX key path value [start [stop]]	`JSON.ARRINDEX employee_profile:1 .colors '"purp`
Description: Searches for the first occurrence of a JSON value in an array. Time Complexity: O(N) wh size of the array, O(N) when path is evaluated to multiple values, where N is the size of the key		

Search and Query

CLI node-redis redis-py NRedisStack Jedis

Command	Syntax

FT.CREATE	`FT.CREATE index`
	`  [ON HASH | JSON]`
	`  [PREFIX count prefix [prefix ...]]`
	`  [FILTER {filter}]`
	`  SCHEMA`
	`  field_name [AS alias] TEXT | TAG | NUMERIC | GEO | VECTOR | GEOS`
	`  [NOINDEX]`
	`   ...`

Description: Create an index with the given specification.Time Complexity: O(K) where K is the numb	

FT.SEARCH	`FT.SEARCH index query`
	`  [FILTER numeric_field min max [ FILTER numeric_field min max ...`
	`  [RETURN count identifier [AS property] [ identifier [AS property`
	`  [SORTBY sortby [ ASC | DESC] [WITHCOUNT]]`
	`  [LIMIT offset num]`
	`  [PARAMS nargs name value [ name value ...]]`

Description: Search the index with a query, returning either documents or just ids. Time Complexity	

FT.AGGREGATE	`FT.AGGREGATE index query`
	`  [LOAD count field [field ...]]`
	`  [ GROUPBY nargs property [property ...] [ REDUCE function nargs`
	`  [ SORTBY nargs [ property ASC | DESC [ property ASC | DESC ...]`
	`  [ APPLY expression AS name  ...`
	`  [ LIMIT offset num]`
	`  [FILTER filter]`
	`  [ PARAMS nargs name value [ name value ...]]`

Description: Run a search query on an index, and perform aggregate transformations on the results	
FT.INFO	FT.INFO index
Description: Return information and statistics on the index.Time Complexity: O(1)	
FT.DROPINDEX	FT.DROPINDEX index [DD]
Description: Dropping existing index.Time Complexity:O(1) or O(N) if documents are deleted, where	

Join Redis University

Get Started with Redis

This path is designed for developers who are new to Redis. Whether you’re ready to start using Redis in production or just