

By OnlineInterviewQuestions.com

Redis Interview Questions

Practice here the top **Redis Interview Questions and Answers**, that are mostly asked during Redis Job Interviews.

Q1. What is Redis?

Redis is an open-source released under BSD licensed, in-memory data structure store, it can be used as a database, cache and message broker.

Q2. List the data structures supported by Redis.

Redis supports following Data Structures

- Strings
- Hashes
- Lists
- Sets
- Sorted sets with range queries
- bitmaps
- Hyperloglogs
- Geospatial indexes with radius queries

Q3. List some advantages of Redis?

Following are some advantages of Redis

- Exceptionally Faster than [others](#)
- Supports for rich data types like Hashes, Sets, bitmaps
- Rich client-side library.
- Support for server-side locking.
- Operations are atomic.

Q4. List some commonly used Redis commands?

Some commonly used Redis commands list

Command Name	Description
APPEND	Append a value to a key
AUTH	Authenticate to the server
BGREWRITEAOF	Asynchronously rewrite the append-only file
BGSAVE	Asynchronously save the dataset to disk
BITCOUNT	Count set bits in a string
CLIENT LIST	Get the list of client connections
CLUSTER INFO	Provides info about Redis Cluster node state

Read more from [Redis Commands List](#)

Q5. In which language Redis is written?

Redis is [NoSql](#) based Key-value Database, which is written in **ANSI C**

Q6. List some Redis Clients supported by PHP?

Below are some Redis Clients supported by [PHP Programming Language](#)

- amphp/redis
- cheprasov/php-redis-client
- Credis
- PHP Redis implementation / wrapper
- PHP Sentinel Client
- phpredis

Q7. What is redis-cli.

redis-cli is the Redis command-line interface, a simple program that allows sending commands to Redis, and read the replies sent by the server, directly from the terminal.

Q8. Explain REPL

REPL stands for Read Eval Print Loop. It an interactive mode where the user types commands and get replies.

Q9. List the programming languages supported by Redis?

Redis supports a wide range of programming language. Some major programming languages supported by Redis are PHP, Java, Python, Scala, Perl, Ruby, C#, and C++.

Q10. What is difference between Redis and Memcached?

The differences between Redis and Memcached are as follows:

REDIS

It was released in 2009.

It was developed by Salvatore Sanfilippo.

It uses single cores.

In Redis, the maximum key length is 2GB.

It is simple and easier to install as compared to Memcached.

It uses list, strings, hashes, sorted sets and bitmaps as data structure.

Its reads and writes speed is slower than Memcached.

It supports Master-Slave Replication and Multi-Master Replication methods.

It is more durable than Memcached.

It has Document Store, Graph DBMS, Search Engine, and Time Series DBMS as secondary database models.

It uses persistent data.

It supports Sharding.

MEMCACHED

It was released in 2003.

It was developed by Danga Interactive.

It uses multiple cores.

In Memcached, maximum key length is 250 bytes.

It is difficult to install.

It uses only string and integers as data structure.

Its reads and writes speed is higher than Redis.

It does not support any replication method.

It is less durable than Redis.

It has no secondary database models.

It does not use persistent data.

It does not support any partitioning method.

Q11. Enlist some operation keys of Redis?

Some operation keys of Redis are listed bellow:

Command

APPEND key value

AUTH password

BGREWRITEAOF

BITPOS key bit [start] [end]

BLPOP key [key ...] timeout

CLIENT LIST

CLIENT PAUSE timeout

CLUSTER FORGET node-id

Description

Append a value to a key

Authenticate to the server

Asynchronously rewrite the append-only file

Find the first-bit set or clear in a string

Remove and get the first element in a list, or block until one is available

Get the list of client connections

Stop processing commands from clients for some time

Remove a node from the nodes table

Command	Description
CLUSTER INFO	Provides info about Redis Cluster node state
CLUSTER KEYSLOT key	Returns the hash slot of the specified key
CLUSTER MEET ip port	Force a node cluster to handshake with another node
CLUSTER NODES	Get Cluster config for the node
CLUSTER SAVECONFIG	Forces the node to save cluster state on disk
CLUSTER SLAVES node-id	List slave nodes of the specified master node
CLUSTER SLOTS	Get an array of Cluster slot to node mappings
COMMAND	Get an array of Redis command details
COMMAND COUNT	Get the total number of Redis commands
COMMAND GETKEYS	Extract keys are given a full Redis command
CONFIG GET parameter	Get the value of a configuration parameter
CONFIG REWRITE	Rewrite the configuration file with the in-memory configuration
CONFIG SET parameter value	Set a configuration parameter to the given value
CONFIG RESETSTAT	Reset the stats returned by INFO
DBSIZE	Return the number of keys in the selected database
DEBUG OBJECT key	Get debugging information about a key
DEBUG SEGFAULT	Make the server crash
DECR key	Decrement the integer value of a key by one
DECRBY key decrement	Decrement the integer value of a key by the given number
DEL key [key ...]	Delete a key
DISCARD	Discard all commands issued after MULTI
DUMP key	Return a serialized version of the value stored at the specified key.
ECHO message	Echo the given string
EXEC	Execute all commands issued after MULTI
EXPIRE key seconds	Set a key's time to live in seconds
GET key	Get the value of a key
GETBIT key offset	Returns the bit value at offset in the string value stored at key
GETRANGE key start end	Get a substring of the string stored at a key
GETSET key value	Set the string value of a key and return its old value
HDEL key field [field ...]	Delete one or more hash fields
HEXISTS key field	Determine if a hash field exists
HGET key field	Get the value of a hash field
HGETALL key	Get all the fields and values in a hash
HKEYS key	Get all the fields in a hash
HLEN key	Get the number of fields in a hash
HSET key field value	Set the string value of a hash field
HSETNX key field value	Set the value of a hash field, only if the field does not exist
HSTRLEN key field	Get the length of the value of a hash field
HVALS key	Get all the values in a hash
INCR key	Increment the integer value of a key by one

Command	Description
INCRBY key increment	Increment the integer value of a key by the given amount
INFO [section]	Get information and statistics about the server
KEYS pattern	Find all keys matching the given pattern
LASTSAVE	Get the UNIX timestamp of the last successful save to a disk
LINDEX key index	Get an element from a list by its index
LLEN key	Get the length of a list
LPOP key	Remove and get the first element in a list
LPUSHX key value	Prepend a value to a list, only if the list exists
LRANGE key start stop	Get a range of elements from a list
LREM key count value	Remove elements from a list
LSET key index value	Set the value of an element in a list by its index
LTRIM key start stop	Trim a list to the specified range
MGET key [key ...]	Get the values of all the given keys
MONITOR	Listen for all requests received by the server in real-time
MOVE key db	Move a key to another database
MULTI	Mark the start of a transaction block
PERSIST key	Remove the expiration from a key
PFCOUNT key [key ...]	Return the approximated cardinality of the set(s) observed by the HyperLogLog at key(s).
PING [message]	Ping the server
PTTL key	Get the time to live for a key in milliseconds
QUIT	Close the connection
RANDOMKEY	Return a random key from the keyspace
READONLY	Enables read queries for a connection to a cluster slave node
READWRITE	Disables read queries for a connection to a cluster slave node
RENAME key newkey	Rename a key
ROLE	Return the role of the instance in the context of replication
RPOP key	Remove and get the last element in a list
RPUSHX key value	Append a value to a list, only if the list exists
SAVE	Synchronously save the dataset to disk
SCARD key	Get the number of members in a set
SCRIPT FLUSH	Remove all the scripts from the script cache.
SCRIPT KILL	Kill the script currently in execution.
SCRIPT LOAD script	Load the specified Lua script into the script cache.
SDIFF key [key ...]	Subtract multiple sets
SELECT index	Change the selected database for the current connection
SETNX key value	Set the value of a key, only if the key does not exist
SINTER key [key ...]	Intersect multiple sets
SLAVEOF host port	Make the server a slave of another instance, or promote it as master
SMEMBERS key	Get all the members in a set
SPOP key [count]	Remove and return one or multiple random members from a set

Command	Description
STRLEN key	Get the length of the value stored in a key
SWAPDB index index	Swaps two Redis databases
SYNC	An internal command used for replication
TIME	Return the current server time
TOUCH key [key ...]	Alters the last access time of a key(s). Returns the number of existing keys specified.
TTL key	Get the time to live for a key
TYPE key	Determine the type stored at key
UNLINK key [key ...]	Delete a key asynchronously in another thread. Otherwise, it is just as DEL, but non-blocking.
UNWATCH	Forget about all watched keys

Q12. What is ZSET in Redis?

ZSET refers to the Redis Sorted Set where a Redis data type documented. In a sorted set, Each key has multiple values inside that is associated with a floating value score. In Redis, It has the unique property of being able to be accessed by a member like a HASH but items can also be accessed by the sorted order and values of the scores.

Q13. What are Redis Hashes?

Redis hashes are defined as the hashes that map string names to string values. They are containers of unique fields and their values. It is a perfect way to represent an object as a Redis data structure. It provides constant time basic operations such as get, set, exists, etc.

Please Visit OnlineInterviewquestions.com to download more pdfs