13 reply

m-combinator 6 months ago

I found this repo to be helpful for this stage.

4 reply

mevimo 22 days ago

For those that wish to implement the CRC checksum validation: beware, Redis does not use the standard CRC64-ECMA or ISO variants, but a special "Jones" variant instead. You can see Redis' implementation (in C) here, with more examples in Rust and in Go.

2 reply

wwarne 4 months ago

In the write-up i saw this: Length Encoding: Read one byte from the stream, compare the two most significant bits. If it's '10' - Discard the remaining 6 bits. The next 4 bytes from the stream represent the length.

But in reality if you have '0b10' bits, you need to make another check - compare the whole first byte: if first byte == 0x80 -> Length of the string is stored as 32 bit integer (Big-Endian encoding!) so you need to read 4 next bytes if first byte == 0x81 -> Length of the string is storead as 64 bit integer (Big-Endian encoding again!) so you need to read 8 next bytes

I checked in different repositories - https://github.com/sripathikrishnan/redis-rdb-tools/ (python), https://github.com/dongmx/rdb/ (go), and https://github.com/redis/librdb/blob/main/src/lib/parser.c and they all have this logic related to string lengths. My guess it's just was added to the standart after the write-up was published.

1 reply