6 0

0

0

### <<

20 从零实现 KV 存储一String 结构支持

Set

Del

# 20 从零实现 KV 存储—String 结构支持

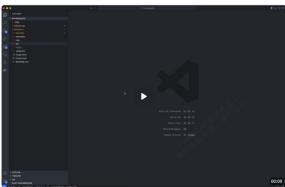
### 理论讲解



## Go 语言编码



## Rust 语言编码



在上一节提到的总体设计中,对于 Redis 的 String 类型,我们设计的编码结构如下:

## type: 数据类型

```
1 0-String
2 1-Mash
3 2-Set
4 3-List
5 4-ZSet
```

expire: 过期时间, unix 时间戳 payload: 原始 value 部分

## Set

Put 一个 key/value 键值对,只需要将 value 加上对应的编码结构即可,然后调用存储引擎的接口,编码的逻辑和我们之前学习过的对 LogRecord 的编码比较类似,利用的库和编码的方法都是一样的。

## Get

候据用户传递的 Key 直找对应的 Value,如果 Value 个存在的话,则说明个存在这个 Key,直接这凹。

否则需要判断类型,如果不是一个 String 类型的 key,则说明类型不匹配,则直接返回一个错误,在 Redis 中,这个错误的说明如下所示:

1 127.8.8.1:6379> get myzset 2 (error) WRONGTYPE Operation against a key holding the wrong kind of value

如果是的话,则说明是对应 key 类型的数据,需要将其解码,拿到过期时间和实际的 value,如果已经过期了,那么也直接返回。

#### Del

Del 是一个通用的命令,可以删除任意类型的 key 数据,我们直接询用存储引擎的接口删除。 如果是非 String 类型的数据结构,其实删除的是它的元数据,因为这种类型的数据我们首先会查元数据,如果元数据都不存在了,那么所属这个 key 的数据都是无效的。

#### Type

Type 也是 Redis 中的一个通用命令,可以获取一个 key 的类型,对于 String 类型,我们的 value 中维护了类型,对于其他的四种数据结构,在元数据中也存储了对应的类型,所以我们直接 Get 数据,然后解码获取类型即可,



