

<<

22 从零实现 KV 存储—Set 结构支持

SADD

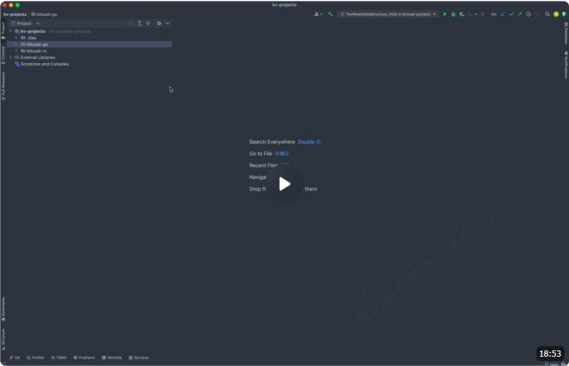
SISMEMBER

SREM

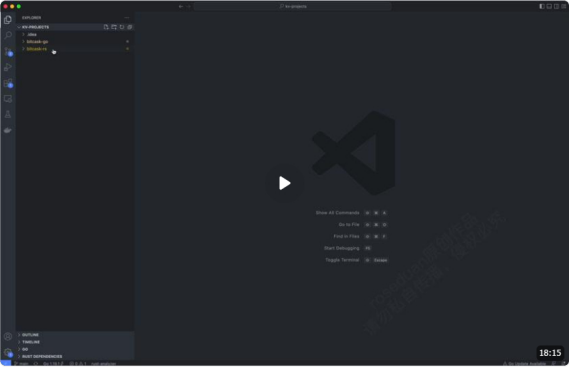
理论讲解



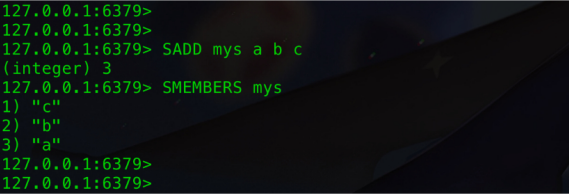
Go 编码



Rust 编码



在前面的 Redis 数据结构总体设计中，我们对于 Set 结构的编码设计如下所示：



元数据

1	key =>	type	expire	version	size
2		(1byte)	(Ebyte)	(8byte)	(Sbyte)
3					
4					

和 Hash 数据结构的设计完全一致。

数据部分

SQL	
1	
2	key version member member size => NULL
3	

这里需要注意我们存储的 value 是空的，member 只需要从 key 里面获取就可以了，因此在编码的时候，可以把 member 的长度编码进去，获取到 key 之后，从最后的部分拿到 member size，即 member 的长度，然后截取这个长度就是 member 的值了。

SADD

先获取元数据，如果不存在的话，初始化一个元数据结构体，然后组合成一个数据部分的 key，并且更新元数据和

数据。

SISMEMBER

和 SAdd 类似。首先获取到元数据，然后构造数据部分的 key，调用存储引擎的接口查看是否存在。

SREM

先获取元数据，如果不存在的话，说明没有这个 key，直接返回。然后也需要构造一个数据部分的 key，调用存储引擎的接口，如果存在的话，则更新元数据，并且删除这个 key。



真诚点赞，手留余香

输入评论

