**Title:** Redis XGROUP

***Excerpt:*** Redis XGROUP command is used to manage the consumer groups associated with a given stream. It supports several subcommands to create and delete consumer groups and consumers.

**Permalink:** redis-xgroup

**Category:** Redis

# 

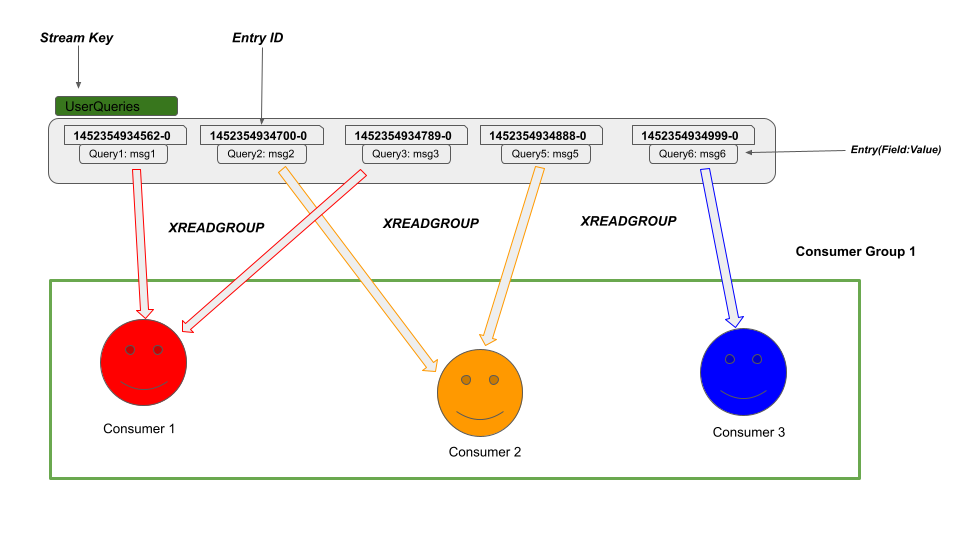
# 

# 

# Redis Streams and Consumer Groups

Redis streams have been introduced from version 5.0 which represents a traditional append-only log data structure. Redis streams are more advanced than a usual log where it supports consumer groups and faster access to the stream elements.

Streams enable the same stream to be consumed by different consumers using the XREAD command. With this, the same messages will be sent to multiple clients. In some cases, we need to push a different subset of messages from the same stream to multiple consumers. The consumer groups concept has been implemented with Redis streams to support this use case as in the following illustration.



As shown in the above illustration, the three consumers *Consumer1, Consumer2, and Consumer3* are consuming different entries from the stream *UserQueries.* Each of these consumers is part of consumer group *1.* Being a member of the ‘Consumer Group 1’ guarantees the following facts.

* Each entry will be consumed by a different consumer out of three consumers at a given time. No entry is consumed by two or more consumers at once.
* An explicit acknowledgment is needed to notify that the pushed messages have been properly consumed by a consumer.
* Consumer Group 1 holds the last delivered entry id and whenever a consumer asks for new messages, it provides the latest messages only.
* Whenever a consumer queries the history of the entries belonging to a given stream, it will only show the messages that are delivered to that particular consumer.
  + *Consumer 2* only sees *Query2:msg2* and Query5:msg5 entries as the history of messages belonging to the *‘UserQueries’* stream.

# The XGROUP Command

The XGROUP is the container command for managing the consumer groups associated with Redis streams. It consists of several important subcommands to create/delete consumer groups and create/delete consumers belonging to a group. The basic syntax of the XGROUP command is as follows.

| XGROUP <subcommand> [<args> [value] [options] ...] |
| --- |

***<subcommand>:*** The subcommand name for operations like creation and deletion of consumers and consumer groups.

***<args>:***The subcommand arguments if available.

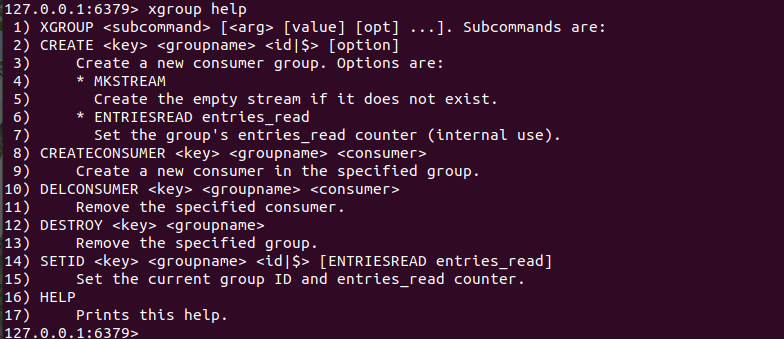
***[value]:*** If an argument takes a value, it can be specified. It is optional and permitted only for certain commands.

The returned result will vary based on the sub-command specified.

## The XGROUP Subcommands

Several subcommands are available to use with the XGROUP command. There is a special subcommand called ‘*help’* which displays all the subcommands available to use.

| XGROUP help |
| --- |



The *help* subcommand returns an array of subcommands with their descriptions.

CREATE

The *CREATE* subcommand can be used to create a new consumer group for a given stream. The syntax is as follows.

| XGROUP CREATE stream\_key group\_name entry\_id|$ [MKSTREAM] [ENTRIESREAD entries\_read] |
| --- |

This will create a new consumer group identified by the ‘group\_name’ that is associated with the given ‘*stream\_key’*. The *‘MKSTREAM’* option creates a new stream with the length of 0 if the specified stream doesn’t exist in the Redis datastore.

CREATECONSUMER

When you need to create a consumer for a given group, the CREATECONSUMER subcommand can be used. It accepts only the stream key, consumer group name, and a unique consumer name as arguments.

| XGROUP CREATECONSUMER stream\_key group\_name consumer\_name |
| --- |

DELCONSUMER

A created consumer can be deleted using the DELCONSUMER subcommand with the XGROUP command. The DELCONSUMER subcommand syntax is as follows.

| XGROUP DELCONSUMER stream\_key group\_name consumer\_name |
| --- |

The most important thing to keep in mind before you delete a consumer from the consumer group is to claim or acknowledge all the pending messages associated with the consumer. This command will return the number of pending messages associated with the deleted consumer.

DESTROY

Whenever you need to delete a consumer group with all consumers and pending messages, the DESTROY subcommand can be used. So, it is recommended to use this command with care.

| XGROUP DESTROY stream\_key group\_name |
| --- |

This command will return an integer 0 or 1 which is the count of the deleted consumer groups.

SETID

The SETID subcommand allows you to modify the last delivered id value. Whenever you need to process all the entries from the stream, the last delivered id should be set to 0 using the SETID subcommand.

| XGROUP SETID stream\_key group\_name entry\_id|$ [ENTRIESREAD entries\_read] |
| --- |

If the last delivered id has been set properly, the command will return a simple string *OK*.

# Conclusion

In short, the XGROUP container command is used to manage the consumer groups associated with a given stream. As discussed, the special *HELP* command can be used to display all the available subcommands for the XGROUP command. Creation and deletion of consumers and consumer groups can be done using the mentioned subcommands. In addition, the SETID subcommand supports updating the last delivered id of a given consumer group. Overall, the XGROUP command output varies with the subcommand being used.