**Title:** Redis SREM

***Excerpt:***Redis sets are a collection of unordered strings which doesn’t contain repeating elements. Usually, the SREM command is used to remove an element from a list. From the Redis version 2.4.0, the SREM command facilitates the removal of multiple elements simultaneously. This command has constant time complexity.

**Permalink:** redis-srem

**Category:** Redis

# 

# 

# 

# Introduction to Redis Sets

If your next web application looks for a way to store a collection of strings without duplicates, Redis sets are the go-to data structure. Redis sets can contain more than four billion string type members which are more than enough in practical use cases. Several pre-built commands are available to perform basic to advanced operations on sets such as addition, removal, existence check, etc. The most important thing about the basic set operations like add and remove is that they have constant time complexity.

# Redis SREM Command

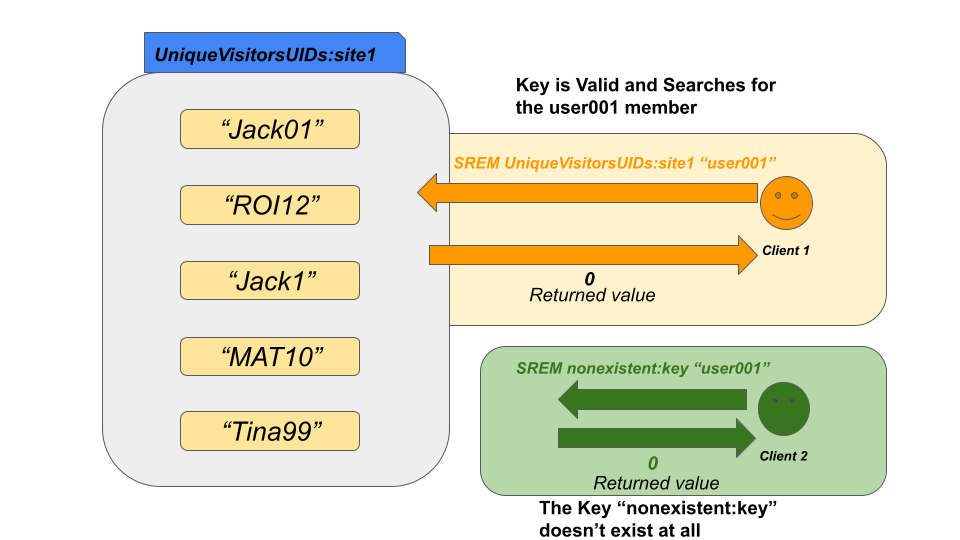
Redis SREM command is used to remove elements from a set stored at a specified key. From version 2.4.0, this command supports removing multiple members simultaneously. Validations are in place for the following scenarios.

* Specifying non-existing members

If a specified member is not a part of the given set, that member will be skipped.

* Specifying non-existing key

Whenever the key is not available, the SREM command will return 0. In that case, the SREM command treats that non-existing set as an empty set.



## Syntax:

The SREM command has very simple syntax as shown in the following.

| SREM redis\_key member [member ...] |
| --- |

***redis\_key***: The unique identifier where the set is stored.

***member***: The member to be removed from the list. Optionally, you can specify multiple members.

The SREM command returns an integer value which is the count of the members removed successfully. As an example, if three members were removed from the set, the returning value would be 3.

In addition, if you specify a key that points to a hash or list but not to a set. In such a case, the SREM command would raise an error.

## Example 01 - Remove a Member From the Set

Let’s assume that we are going to store top fans of your personal website based on the interactivity level each month. Since sets keep non-duplicate members, we can assure that the same user will not be added multiple times.

*We will be using the SADD command to add the top 5 fans in January to the set stored at key topfans:january.*

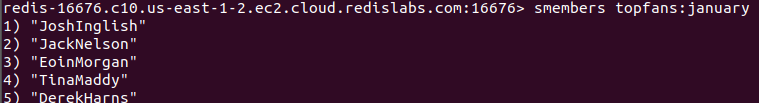
| sadd topfans:january JackNelson TinaMaddy JoshInglish EoinMorgan DerekHarns |
| --- |



Let’s inspect the *topfans:january* set using SMEMBERS command.

| smembers topfans:january |
| --- |

Output:

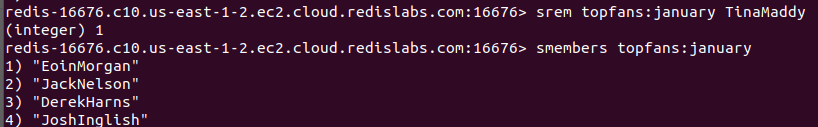


Due to the unethical behavior of the member “TInaMaddy”, we got a requirement to remove her from the set. Now the SREM command is the ideal candidate to do this.

| srem topfans:january TinaMaddy |
| --- |

As expected the output is 1 which means one member has been removed. Let’s inspect the existing members in the set as well.

| smembers topfans:january |
| --- |



As shown in the above output, the member “TinaMaddy” has gone from the list.

## Example 02 - Remove Multiple Members Simultaneously

In some scenarios, we might need to remove more than one member from a set. Hence, the SREM command supports multiple member arguments. Let’s say we need to remove both the members “JackNelson” and “JoshInglish”. The following command would do that for you.

| srem topfans:january JackNelson JoshInglish |
| --- |



Command returned the integer 2 which means two members have been removed.



The important thing to notice is that the SREM command consumes constant time in both scenarios. Hence, it is recommended to use in low latency applications.

# Conclusion

To summarize, the SREM command operates on set data structures. Its main job is to remove an element from a set stored at a specified key. Like the SADD and SISMEMBER commands, the SREM command has constant time complexity where the number of elements will not take effect on the execution time of the SREM command. As stated, from the Redis version 2.4.0, the SREM command allows multiple member arguments. Hence, you can delete multiple members from a given list simultaneously. Upon the execution of this command, it returns an integer value which is the number of successfully removed elements from the list. Overall, the SREM command is an ideal candidate to use in low latency applications.