**Title:** Redis LSET

***Excerpt:*** The LSET command is used to update or set an element at the specified index for a given list stored at a specified key. This command has constant time complexity when setting elements at head and tail. Whenever a list element is accessed from within the list, it has O(N) time complexity. The LSET command returns a simple string 'OK' If the list element is successfully set. An error will be returned if the element index doesn't exist.

**Permalink:** redis-lset

**Category:** Redis

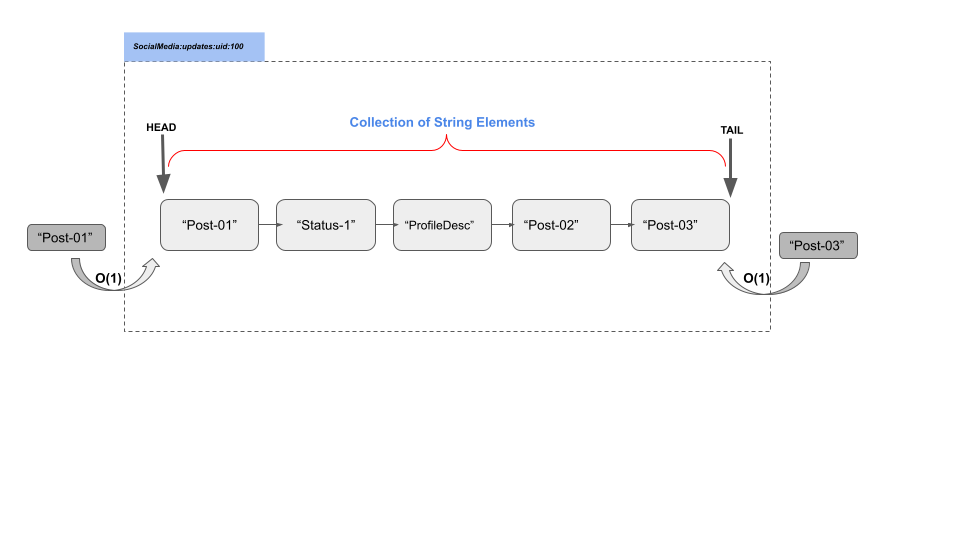
# 

# 

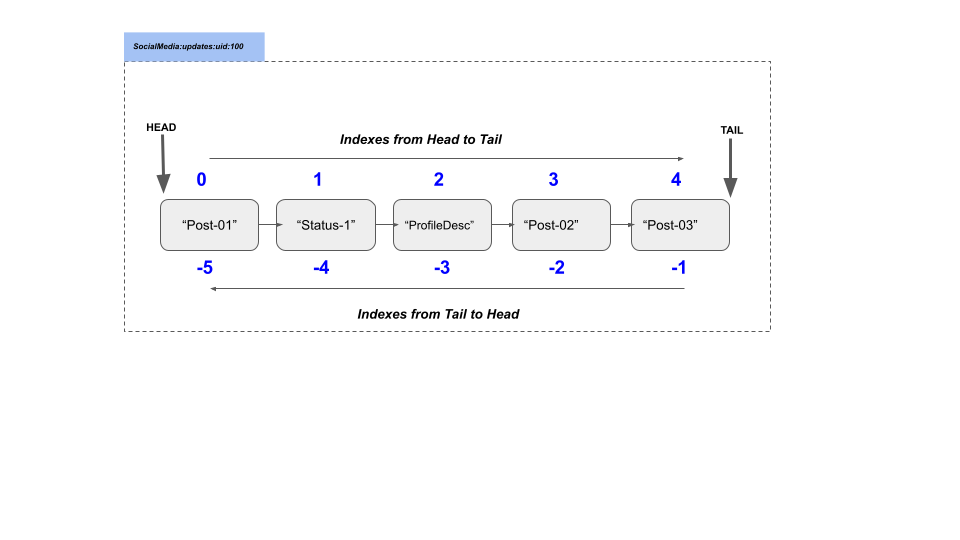
# 

# Properties of Redis Lists

A Redis list is a data structure that holds a sequence of string elements in the insertion order. It is based on Linked List. So, the insertion of an element at the head and tail is performed in constant time. The only downside of the Redis lists is that querying elements by an index is a bit costly which takes time directly proportional to the index of the accessing element. The reason behind the Linked list-based implementation is to make the data insertion faster for a very long list of items.



Redis lists have 0-based indexing. The first element is designated as the 0th index, the second one is at the 1st index, and so on. Accessing tail elements with the indexes is possible with negative indices like the last element is indexed as -1, the penultimate element is -2, and so on.



The list elements can be set at any given index with the LSET command which is described in the following section.

# The LSET Command

The LSET command sets an element at the given index of a Redis list stored at the specified key. It has a very simple syntax as follows.

| LSET list\_key index element |
| --- |

***list\_key***: The key of the Redis list.

***index***: The index where you need to set an element.

***element***: The string element to be set at the specified index.

This command returns a Simple string “OK” if the element has been set properly. Whenever it encounters an index that is out of the range, the LSET command returns an error.

One of the most common use cases of the Redis lists is to store social media posts and their updates for individual users. The LSET command plays an important role there as discussed in the following section.

# Use Case - Update Social Media Posts

Let’s assume an event where a company maintains its own social media website with thousands of users actively posting posts and statuses. They have identified that the Redis lists are the suitable candidate to hold these posts and statuses where frequent insertion happens and the Redis lists perform insertion in constant time.

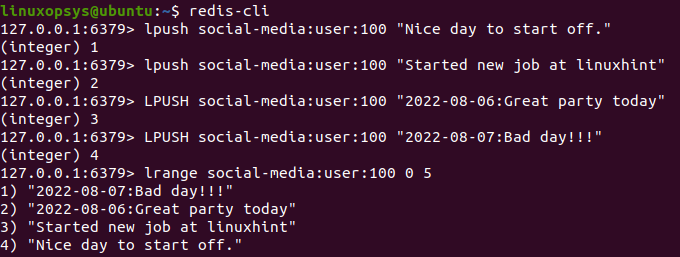
Let’s create a Redis list for a given user id 100 that will be used to store that particular user’s posts and status messages. We will be using the LPUSH command to put some dummy posts and status messages for the user id 100.

| LPUSH social-media:user:100 "2022-08-01:Nice day to start off."  LPUSH social-media:user:100 "2022-08-05:Started new job at linuxhint"  LPUSH social-media:user:100 "2022-08-06:Great party today"  LPUSH social-media:user:100 "2022-08-07:Bad day!!!" |
| --- |

Let’s check what the list looks like using the LRANGE command.

| LRANGE social-media:user:100 0 5 |
| --- |

Output:



Assume that the user with user id 100 is going to update one of his old posts. Let’s say the third post that he has posted needs some correction. This can be easily achieved by the LSET command as shown in the following.

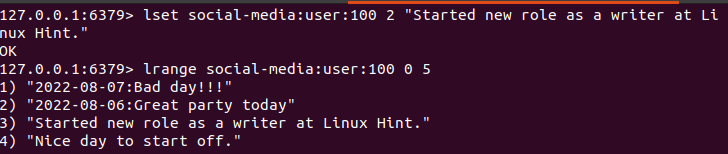
Since the third element is at index 2 from the Head. We will be updating it as follows.

| LSET social-media:user:100 2 "Started new role as a writer at LinuxHint" |
| --- |

Output:

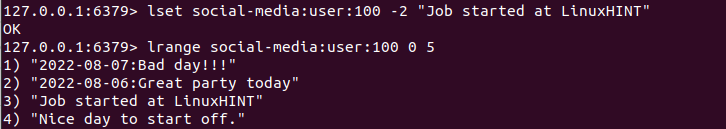


As expected, the command was executed successfully and the element at the second index was updated. Let’s check the list again as follows.



We can do the same by specifying the index from the tail as follows.

| LSET social-media:user:100 -2 "Started new role as a writer at LinuxHint" |
| --- |



As expected, the third element has been set to the new string “Job started at LinuxHINT”.

If you specify an index that is out of range, it would raise an error as shown in the following.

| LSET social-media:user:100 6 "Job started at LinuxHINT" |
| --- |

Output:



# Conclusion

In short, the LSET command sets a list element at a specified index to a given element of the Redis list stored at a given key. Since the Redis list indexes start from 0, it is the first element. Index 1 is the second element of the list and so on. As discussed, the indexes of the elements starting from the tail can be specified using negative numbers like -1 is the last element, -2 is the element before the last, and so on. Whenever a non-existing index is passed, the LSET command will return an error. Furthermore, this command returns a simple string output over the successful execution.