



delete/all= used to delete the all keys and values present in the go clients  
 delete/key = used to delete the particular key and value pair from the go client  
 post/key/value/time = used to store the key value pair with respective the time in the go clients  
 get/inmemory/print = used to print the key values pairs present in the in memory  
 get/redis/print = used to print the key values pairs present in the redis  
 get/key= used to get the value for the particular key and compare between redis and in memory , if both are same print the value or nothing is printed.

## REDIS

### Del\_all

Use the `FlushAll` method to delete all key-value pairs in the Redis database.

Print:

Use the `LRange` method to get all keys present in the `cache` list.

For each key, check if the value is present in Redis.

If the value is not present, delete the key from the `cache` list.

At the end, return a string representation containing the key-value pairs.

DEI:

Check if the specific key exists.

If it exists, remove the key from the `cache` list and delete the key-value pair from Redis.

Get:

Use the `Get` function to retrieve the value for the given key.

If an error occurs or if the key does not exist, return an empty string.

If the key is found, remove it from the `cache` list and add it to the front.

Return the retrieved value.

Put:

Check if the key already exists in Redis.

If it exists, remove the key from the `cache` list and add it to the front, then update the value.

If it does not exist:

- Compare the length of the `cache` list to the maximum length.
- If the length is less than the maximum, add the new key-value pair.
- If the length is more, remove the least recently used item and add the new key-value pair.

---

---

## INMEMORY

Delete:

Deletes a specific key-value pair from the cache if it exists.

Removes the element from both the map and the doubly linked list.

Get:

Retrieves the value associated with the given key.

Checks if the key has expired; if not, moves the accessed element to the front of the list and returns its value.

If the key is expired, remove it from the cache and return an empty string.

Put:

Adds or updates a key-value pair in the cache with an optional TTL.

If the key already exists, updates its value and TTL, and moves it to the front of the list.

If the cache exceeds `maxLength`, evicts the least recently used element before adding the new key-value pair.

`Del_all`

clears the entire cache by reinitializing the doubly linked list and resetting the map that holds cache entries.

`Print:`

Returns a string representation of the cache contents in order from most to least recently used.

Iterates from the start of the list, checking the TTL for each key.

Deletes expired keys and constructs a string of the remaining key-value pairs.