

# Multi-Backend Caching Library in Go

## Link:

[https://github.com/Devisree146/Multi\\_Backend\\_Caching\\_In\\_Golang.git](https://github.com/Devisree146/Multi_Backend_Caching_In_Golang.git)

Create a new Go\_project, with a go.mod file and a go.sum file.

**\*\* go mod init**—For go.mod file

**\*\* go mod tidy**—For go.sum file

## IN-MEMORY CACHE

These functions collectively manage an in-memory cache with LRU eviction policy, ensuring thread-safe operations and periodic cleanup of expired items.

Methods for in\_memory:

**Set:** Adds or updates an item in the cache. Moves updated items to the front of the LRU list and evicts items if the capacity is exceeded.

**Get:** Retrieves an item from the cache if it exists and hasn't expired. Updates the item's position in the LRU list.

**Deletekey:** Internally deletes an item from the cache without locking. Used in the eviction routine and **Get** method.

**GetAll:** Retrieves all non-expired items from the cache. Initiates eviction for expired items.

**Delete:** Removes a specific item from the cache if it exists.

**DeleteAll:** Clears all items from the cache.

**evict:** Removes the least recently used (LRU) item from the cache when capacity is exceeded.

## REDIS CACHE

These functions collectively provide a Redis-based cache with LRU eviction policy and CRUD operations.

Methods for redis caching:

**Set:** Adds a key-value pair to the cache with a specified time-to-live (TTL).

Updates the

access order for LRU and evicts elements if necessary.

**Get:** Retrieves the value for a given key from the cache and updates the access order for

LRU.

**GetAll:** Retrieves all key-value pairs currently stored in the cache.

**Delete:** Removes a specific key from the cache and the LRU list.

**DeleteAll:** Clears the entire cache by deleting all keys and the LRU list.

## UNIFIED API:

Methods in Unified API:

**SET:** **POST /cache/**

**Description:** Sets a key-value pair in the cache with an expiration time.

**Parameters:** JSON object containing **key**, **value**, and **expiration** (in seconds).

**Response:** Returns a success message upon setting the value.

**GET:** **GET /cache ?key="Key value"**

**Description:** Retrieves the value associated with the specified key from the cache.

**Parameters:** **key** – The key to look up in the cache.

**Response:** Returns the value if the key is found; otherwise, returns a 404 error.

**GETALL:** **GET /cache/all**

**Description:** Retrieves all key-value pairs from the cache.

**Response:** Returns a JSON object containing all items in the cache.

**DELETE:** **DELETE /cache ?key="key value"**

**Description:** Deletes the specified key from the cache.

**Parameters:** **key** – The key to delete from the cache.

**Response:** Returns a success message if the key is deleted; otherwise, returns a 404 error.

**DELETEALL:** **DELETE /cache/all**

**Description:** Deletes all keys from the cache.

**Response:** Returns a success message if all keys are deleted; otherwise, returns a 404

error if no keys are found.

The API interaction can be done with Postman or using CURL commands as well.

## BENCHMARK RESULTS:

```
directory not found
PS C:\Users\devi\OneDrive\Desktop\python-sample-tests\Go_project\test\multicache_test> go test -bench=. -bench
time=5s -v
=== RUN TestCacheAPI
[GIN-debug] [WARNING] Creating an Engine instance with the Logger and Recovery middleware already attached.

[GIN-debug] [WARNING] Running in "debug" mode. Switch to "release" mode in production.
- using env:   export GIN_MODE=release
- using code:  gin.SetMode(gin.ReleaseMode)

[GIN-debug] POST   /cache          --> unified/multicache.SetupRouter.func1 (3 handlers)
[GIN-debug] GET    /cache          --> unified/multicache.SetupRouter.func2 (3 handlers)
[GIN-debug] DELETE /cache          --> unified/multicache.handleDelete (3 handlers)
[GIN-debug] DELETE /cache/all      --> unified/multicache.SetupRouter.func3 (3 handlers)
[GIN-debug] GET    /cache/all      --> unified/multicache.SetupRouter.func4 (3 handlers)
=== RUN TestCacheAPI/POST_/cache
[GIN] 2024/07/02 - 08:53:53 | 201 |      16.1213ms |      POST   "/cache"
=== RUN TestCacheAPI/GET_/cache
[GIN] 2024/07/02 - 08:53:53 | 200 |           0s |      GET    "/cache?key=test_key"
=== RUN TestCacheAPI/DELETE_/cache/all
[GIN] 2024/07/02 - 08:53:53 | 200 |           0s |      GET    "/cache/all"
[GIN] 2024/07/02 - 08:53:53 | 200 |      520.7µs |     DELETE  "/cache/all"
[GIN] 2024/07/02 - 08:53:53 | 200 |           0s |      GET    "/cache/all"
=== RUN TestCacheAPI/GET_/cache/all_after_deletion
[GIN] 2024/07/02 - 08:53:53 | 200 |      287.5µs |      GET    "/cache/all"
--- PASS: TestCacheAPI (0.13s)
    --- PASS: TestCacheAPI/POST_/cache (0.02s)
    --- PASS: TestCacheAPI/GET_/cache (0.00s)
    --- PASS: TestCacheAPI/DELETE_/cache/all (0.11s)
    --- PASS: TestCacheAPI/GET_/cache/all_after_deletion (0.00s)
PASS
ok      unified/test/multicache_test    0.393s
```