Create a single Express-based app that serves two separate, unrelated pieces of functionality.

In-memory stack (LIFO):

The first piece of functionality is an in-memory stack (LIFO). This portion of the application should have two endpoints:

- an endpoint to add an item to the stack
- an endpoint to return the top item of the stack
    - requesting an item from the stack should also remove that item from the top of the stack

This process should be in-memory, so you don't need to worry about persisting the stack across restarts of the application.

Example requests:

1. Add "Hello" to stack
2. Add "World" to stack
3. Get item from stack
    a. "World" would be returned
4. Add "Again" to stack
5. Get item from stack
    a. "Again" would be returned
6. Get item from stack
    a. "Hello" would be returned

In-memory key-value store with TTL:

The second piece of functionality is an in-memory key-value store that supports TTLs on the keys.

Your interface should support:

- adding a key to the store
    - setting a TTL should be optional to the client when adding the key
- getting the value for a key
    - this should respect the TTL for the key if provided
- deleting the value stored for a given key

For example:

1. Set "name" to "John"
2. Get "name"
    a. This returns John
3. Get "age"
    a. This returns an empty value
4. Set "name" to "Larry" with a TTL of 30 seconds
5. Get "name" (within 30 seconds of the set)
    a. This returns "Larry"
6. Get "name" (more than 30 seconds after the set)
    a. This returns an empty value

Delivery and Expectations:

In total your Express app should have 5 routes:

- Add to stack
- Get from stack
- Add to key-value store
- Get from key-value store
- Delete from key-value store

Please also include unit tests for this functionality.

Upon receipt of the archive we will run:

```
tar -xf [filename.tar]
npm install
npm test
npm start
```

Any other relevant details to use the service should be included in the README.md file at the root of the archive.

No labels