

## **Project: "Persistent Key-Value Store with Write-Ahead Log and SSTable Implementation"**

### **Introduction:**

The code implements a persistent key-value store utilizing a Write-Ahead Log (WAL) for durability and Sorted String Table (SSTable) files for data storage. It features an HTTP server providing endpoints for setting, deleting, and retrieving key-value pairs, with graceful shutdown capabilities using [LSM tree](#) model.

### **Architecture Overview:**

- **MemDB (In-Memory Database):**
  - Manages in-memory storage of key-value pairs.
  - Utilizes periodic flushing to SSTable files based on a set interval or specific conditions.
- **Write-Ahead Log (WAL):**
  - Handles logging operations (Set, Delete) to ensure durability before data flush.
  - Appends operations as binary records to a log file for recovery and data consistency.
- **SSTable (Sorted String Table):**
  - Represents immutable, sorted data files stored on disk.
  - Implements functions for creating, merging, and compacting SST files for efficient storage.

### **Core Components:**

- **HTTP Endpoints:**
  - **/set:** Sets key-value pairs.
  - **/del:** Deletes key-value pairs.
  - **/get:** Retrieves values for given keys.
  - **/shutdown:** Facilitates graceful server shutdown.
- **Periodic Operations:**
  - Periodic flushing of data from memory to SST files.
  - Periodic compaction of SST files to manage file count and optimize storage.
- **File Handling:**
  - Reading and writing SST files in a binary format.
  - Managing WAL file for logging operations.

### **Concurrency and Data Consistency:**

- Utilizes synchronization mechanisms (mutex) for concurrent access to the in-memory database.
- Implements data consistency checks through checksums to maintain file integrity.

### **How to test the code:**

#### **SET:**

```
curl -X POST "http://localhost:8080/set?key=key1&value=value1"
```

In case you want to make space in the value:

```
curl -X POST "http://localhost:8080/set?key=hi&value=val%20smile"
```

#### **GET:**

```
curl -X GET "http://localhost:8080/get?key=key1"
```

#### **DELETE:**

```
curl -X DELETE "http://localhost:8080/del?key=key1"
```

#### **Shutdown the server:**

It's mandatory because it will display important messages.

```
curl -X GET http://localhost:8080/shutdown
```

### **Conclusion:**

The codebase presents a robust, persistent key-value store with durability and disk-based storage capabilities. It leverages a WAL for logging operations and SSTable files for efficient disk storage, providing an HTTP interface for key-value manipulation.

#### **Note:**

The summary encapsulates the main functionalities and core components. Detailed explanations of specific algorithms, data structures, and HTTP request handling are not included due to space limitations. Further exploration into specific code sections may be required for deeper understanding.

### **References:**

- Source code files: **main.go**, **sst.go**, **memdb.go**, **wal.go** **db\_test.go**
- Additional comments and inline documentation provided in the code.