My Project

Generated by Doxygen 1.13.2

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 3

Class Documentation

3.1 RESPProtocol Class Reference

Static Public Member Functions

- static std::string **encodeSimpleString** (const std::string &str)
- static std::string **encodeBulkString** (const std::string &str)
- static std::string **encodeError** (const std::string &msg)
- static std::string **encodeInteger** (int num)
- static std::vector< std::string > **decodeArray** (const std::string &resp)

The documentation for this class was generated from the following files:

- src/resp_protocol.h
- src/resp_protocol.cpp

3.2 StorageEngine Class Reference

Public Member Functions

- void **set** (const std::string &key, const std::string &value)
- std::string **get** (const std::string &key)
- void **del** (const std::string &key)

The documentation for this class was generated from the following files:

- src/storage_engine.h
- src/storage_engine.cpp

3.3 TCPServer Class Reference

Public Member Functions

- **TCPServer** (int port)
- void **run** ()

The documentation for this class was generated from the following files:

- src/tcp_server.h
- src/tcp_server.cpp

Chapter 4

File Documentation

4.1 resp_protocol.h

```
00001 #ifndef RESP_PROTOCOL_H
00002 #define RESP_PROTOCOL_H
00003
00004 #include <string>
00005 #include <vector>
00006
00007 class RESPProtocol {
00008 public:
00009     static std::string encodeSimpleString(const std::string& str);
00010     static std::string encodeBulkString(const std::string& str);
00011     static std::string encodeError(const std::string& msg);
00012     static std::string encodeInteger(int num);
00013     static std::vector<std::string> decodeArray(const std::string& resp);
00014 };
00015
00016 #endif
```

4.2 storage_engine.h

```
00001 #ifndef STORAGE_ENGINE_H
00002 #define STORAGE_ENGINE_H
00003
00004 #include <unordered_map>
00005 #include <string>
00006 #include <mutex>
00007
00008 class StorageEngine {
00009 private:
00010     std::unordered_map<std::string, std::string> db;
00011     std::mutex db_mutex;
00012
00013 public:
00014     void set(const std::string& key, const std::string& value);
00015     std::string get(const std::string& key);
00016     void del(const std::string& key);
00017 };
00018
00019 #endif
```

4.3 tcp_server.h

```
00001 #ifndef TCP_SERVER_H
00002 #define TCP_SERVER_H
00003
00004 #include <netinet/in.h>
00005 #include <unordered_map>
00006 #include <string>
00007
```

```
00008 #ifdef __linux__
00009     #include <sys/epoll.h>
00010 #elif __APPLE__
00011     #include <sys/event.h>
00012 #endif
00013
00014 class TCPServer {
00015 private:
00016     int server_fd;
00017     struct sockaddr_in address;
00018
00019 #ifdef __linux__
00020     int epoll_fd;
00021 #elif __APPLE__
00022     int kqueue_fd;
00023 #endif
00024
00025     std::unordered_map<std::string, std::string> database;  // Simple key-value store
00026
00027     void add_to_event_loop(int fd);
00028     void accept_new_connection();
00029     void handle_client(int client_fd);
00030     std::string process_command(const std::string &request);
00031
00032 public:
00033     explicit TCPServer(int port);
00034     ~TCPServer();
00035     void run();
00036 };
00037
00038 #endif
```

Index