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

- **StorageEngine** (size_t cap=1000)
- void **set** (const std::string &key, const std::string &value)
- std::string **get** (const std::string &key)
- void **del** (const std::string &key)
- void **flush** ()

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, size_t storage_capacity=1000)
- 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

```
#ifndef RESP_PROTOCOL_H
#define RESP_PROTOCOL_H

#include <string>
#include <vector>

class RESPProtocol {
public:
    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);
};

#endif
```

4.2 storage_engine.h

```
#ifndef STORAGE_ENGINE_H
#define STORAGE_ENGINE_H

#include <unordered_map>
#include <list>
#include <string>

class StorageEngine {
private:
    std::unordered_map<std::string, std::string> kv_store;
    std::list<std::string> lru_list;
    std::unordered_map<std::string, std::list<std::string>::iterator> lru_map;
    size_t capacity;

    void evictIfNeeded();

public:
    StorageEngine(size_t cap = 1000);
    void set(const std::string& key, const std::string& value);
    std::string get(const std::string& key);
    void del(const std::string& key);
    void flush();
};

#endif
```

4.3 tcp_server.h

```
#ifndef TCP_SERVER_H
```

```
#define TCP_SERVER_H

#include <netinet/in.h>
#include <string>
#include "storage_engine.h"

#ifdef __linux__
    #include <sys/epoll.h>
#elif __APPLE__
    #include <sys/event.h>
#endif

class TCPServer {
private:
    int server_fd;
    struct sockaddr_in address;
    StorageEngine storage;

#ifdef __linux__
    int epoll_fd;
#elif __APPLE__
    int kqueue_fd;
#endif

    void add_to_event_loop(int fd);
    void accept_new_connection();
    void handle_client(int client_fd);
    std::string process_command(const std::string &request);

public:
    explicit TCPServer(int port, size_t storage_capacity = 1000);
    ~TCPServer();
    void run();
};

#endif
```

Index