C++ server

0.1

Generated by Doxygen 1.8.15

Contents

Chapter 1

Introduction

This program launches the server on the specified port on the header file (localhost:5000).

This server accepts commands such as:

question#n

(for n in (0, max)) it returns a JSON string containing the question(string, type, options)

Chapter 2

Installation

cd into this folder

type:

make clean

make

To run the server type:

∗∗./server∗∗

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

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

Chapter 5

Class Documentation

5.1 Data Struct Reference

all transmitted info == 1 data

```
#include <server.h>
```

Public Attributes

- long size

  the number of pieces of content
- char type [8]

  The type of transmission requested.
- Byte ∗ content

  Actual content.

5.1.1 Detailed Description

all transmitted info == 1 data

The documentation for this struct was generated from the following file:

- server.h

5.2 Question Class Reference

Public Member Functions

- **Question** (char, std::string, std::vector< std::string >)
- char **getType** ()
- std::string **getQString** ()
- std::vector< std::string > **getOptions** ()

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

- QuestionBank.h
- QuestionBank.cpp

5.3 QuestionBank Class Reference

Public Member Functions

- **QuestionBank** (std::string)
- std::string **getFileName** ()
- Question **getQuestion** (int)
- int **getSize** ()
- int **setFile** (std::string)

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

- QuestionBank.h
- QuestionBank.cpp

5.4 Server Class Reference

Server object.

```
#include <server.h>
```

Public Member Functions

- **Server** (int)
- void init ()

  Initializes Server with openSSL context :0.
- int run ()

  Runs the Server ♡.
- int getPort ()

  Gets port that the server is running on.
- void setCtx (SSL_CTX ∗)

  Sets the SSL_CCONTEXT.
- SSL_CTX ∗ getCtx ()

  Gets the SSL_CTX.
- void setMainLoop (std::function< void(SSL_CTX ∗, int)>)

  Sets the main loop of the server.

5.4.1 Detailed Description

Server object.

This object looks after running the Server, it has been made modular so it's easy to implement once the core is designed

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

- server.h
- serverImp.cpp

Chapter 6

File Documentation

6.1 server.h File Reference

```
#include "globals.h"
#include <iostream>
#include <string.h>
#include <cmath>
#include <regex>
#include <signal.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <openssl/ssl.h>
#include <openssl/err.h>
#include "QuestionBank.h"
```

Classes

- struct Data

  all transmitted info == 1 data
- class Server

  Server object.

Macros

- #define PORT 5000

  The default port the Server runs on if no port specified.
- #define BUFFER_STD 4

  The size of 1 piece of content.

Typedefs

- typedef char Byte

  Defines a Byte through a char.
- typedef struct Data Data

  all transmitted info == 1 data

Functions

- void **error** (const char ∗msg)
- void **action** (int sock)
- void **action2** (int sock)
- void **init_openssl** ()
- void **cleanup_openssl** ()
- SSL_CTX ∗ **create_context** ()
- void **configure_context** (SSL_CTX ∗ctx)

6.1.1 Macro Definition Documentation

6.1.1.1 BUFFER_STD

```
#define BUFFER_STD 4
```

The size of 1 piece of content.

The content needs to be split into pieces of size BUFFER_STD for the server to work properly

6.1.2 Typedef Documentation

6.1.2.1 Byte

```
typedef char Byte
```

Defines a Byte through a char.

1 byte == 1 char

Index