



C++ server

0.1

Launch a C++ server on your computer

Contents

0.1	Installation	1
0.2	Introduction	1
0.3	Class Index	1
0.3.1	Class List	1
0.4	File Index	2
0.4.1	File List	2
0.5	Class Documentation	2
0.5.1	Data Struct Reference	2
0.5.1.1	Detailed Description	2
0.5.2	Question Class Reference	2
0.5.3	QuestionBank Class Reference	3
0.5.4	Server Class Reference	3
0.5.4.1	Detailed Description	3
0.5.4.2	Member Function Documentation	3
0.6	File Documentation	4
0.6.1	server.h File Reference	4
0.6.1.1	Macro Definition Documentation	5
0.6.1.2	Typedef Documentation	5
0.6.2	ssl.h File Reference	6
	Index	7

0.1 Installation

cd into this folder

type:

make clean

make

To run the server type:

****./server****

OpenSSL

Linux

```
sudo apt-get install openssl
sudo apt-get install libssl-dev
```

- This installs openssl 1.0.x when last checked. Some methods might not be compatible

0.2 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)

0.3 Class Index

0.3.1 Class List

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

Data		
	All transmitted info == 1 data	2
Question	2
QuestionBank	3
Server		
	Server object	3

0.4 File Index

0.4.1 File List

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

globals.h	??
QuestionBank.h	??
server.h	4
ssl.h	6

0.5 Class Documentation

0.5.1 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.

0.5.1.1 Detailed Description

all transmitted info == 1 data

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

- [server.h](#)

0.5.2 Question Class Reference

Public Member Functions

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

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

- QuestionBank.h
- QuestionBank.cpp

0.5.3 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

0.5.4 Server Class Reference

[Server](#) object.

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

Public Member Functions

- **Server** (int)
- void [initOpenSSL](#) ()
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 *ctx, int server_fd, int client_fd)>)
Sets the main loop of the server.

0.5.4.1 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

0.5.4.2 Member Function Documentation

0.5.4.2.1 initOpenSSL()

```
void Server::initOpenSSL ( )
```

Initializes [Server](#) with openSSL context :0.

Sets the mainLoop to a loop that runs SSL

0.5.4.2.2 run()

```
int Server::run ( )
```

Runs the [Server](#) ♡.

Starts server, begins accepting client takes care of creation/destruction of sockets Runs the main loop on the sockets

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

- [server.h](#)
- [serverImp.cpp](#)

0.6 File Documentation

0.6.1 server.h File Reference

```
#include "globals.h"
#include "ssl.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 "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 defaultDataLoop (SSL_CTX *ctx, int server_fd, int client_fd)`
This loop runs when no loop is specified.
- `void defaultSSLDataLoop (SSL_CTX *ctx, int server_fd, int client_fd)`
This loop runs when SSL is enabled.
- `void error (const char *msg)`
Error Handling.
- `void action (int sock)`
- `void actionSSL (SSL *ssl)`

0.6.1.1 Macro Definition Documentation

0.6.1.1.1 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

0.6.1.2 Typedef Documentation

0.6.1.2.1 Byte

```
typedef char Byte
```

Defines a Byte through a char.

1 byte == 1 char

0.6.2 ssl.h File Reference

```
#include <openssl/ssl.h>
#include <openssl/err.h>
```

Functions

- void [init_openssl](#) ()
Initializes openSSL.
- void [cleanup_openssl](#) ()
Cleans up openSSL.
- SSL_CTX * [create_context](#) ()
Creates context.
- void [configure_context](#) (SSL_CTX *ctx)
Configures context.

Index

- BUFFER_STD
 - server.h, [5](#)
- Byte
 - server.h, [5](#)
- Data, [2](#)
- initOpenSSL
 - Server, [3](#)
- Question, [2](#)
- QuestionBank, [3](#)
- run
 - Server, [4](#)
- Server, [3](#)
 - initOpenSSL, [3](#)
 - run, [4](#)
- server.h, [4](#)
 - BUFFER_STD, [5](#)
 - Byte, [5](#)
- ssl.h, [6](#)