

## Server Side

Generated by Doxygen 1.8.14



# Contents

<b>1</b>	<b>File Index</b>	<b>1</b>
1.1	File List . . . . .	1
<b>2</b>	<b>File Documentation</b>	<b>3</b>
2.1	HTTPServer.c File Reference . . . . .	3
2.1.1	Macro Definition Documentation . . . . .	4
2.1.1.1	CLIENTS_IN_QUEUE . . . . .	4
2.1.1.2	CONNECTION_TIME_OUT_Sec . . . . .	4
2.1.1.3	CONNECTION_TIME_OUT_USec . . . . .	4
2.1.1.4	INT_SIZE . . . . .	4
2.1.1.5	RCVBUFSIZE . . . . .	4
2.1.1.6	SHMOBJ_PATH . . . . .	4
2.1.2	Function Documentation . . . . .	4
2.1.2.1	getResponse() . . . . .	4
2.1.2.2	interruptHandler() . . . . .	5
2.1.2.3	main() . . . . .	5
2.1.2.4	postResponse() . . . . .	5
2.1.2.5	print() . . . . .	6
2.1.2.6	responseForClient() . . . . .	6
2.1.2.7	sendBytes() . . . . .	6
2.1.2.8	sendMessageToClient() . . . . .	6
	<b>Index</b>	<b>9</b>



# Chapter 1

## File Index

### 1.1 File List

Here is a list of all files with brief descriptions:

<a href="#">HTTPServer.c</a> . . . . .	3
--	---



## Chapter 2

# File Documentation

### 2.1 HTTPServer.c File Reference

```
#include <stdio.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/mman.h>
#include <sys/time.h>
#include <sys/sendfile.h>
#include <fcntl.h>
#include <signal.h>
#include "Die_with_error.c"
```

#### Macros

- #define [CLIENTS\\_IN\\_QUEUE](#) 10
- #define [RCVBUFSIZE](#) 10240
- #define [CONNECTION\\_TIME\\_OUT\\_Sec](#) 3
- #define [CONNECTION\\_TIME\\_OUT\\_USec](#) 3000000
- #define [SHMOBJ\\_PATH](#) "/ipc\_obj"
- #define [INT\\_SIZE](#) 4

#### Functions

- void [interruptHandler](#) (int signal)
- void [responseForClient](#) (int sckt)
- void [sendMessageToClient](#) (int sckt, int code, int file\_size)
- void [postResponse](#) (int sckt, char \*filename, int file\_size)
- void [print](#) (char \*str)
- void [getResponse](#) (int sckt, char \*filename, char \*fileType)
- void [sendBytes](#) (int sckt, int fileSize, int filed)
- int [main](#) (int argc, char \*argv[ ])

## 2.1.1 Macro Definition Documentation

### 2.1.1.1 CLIENTS\_IN\_QUEUE

```
#define CLIENTS_IN_QUEUE 10
```

### 2.1.1.2 CONNECTION\_TIME\_OUT\_Sec

```
#define CONNECTION_TIME_OUT_Sec 3
```

### 2.1.1.3 CONNECTION\_TIME\_OUT\_USec

```
#define CONNECTION_TIME_OUT_USec 3000000
```

### 2.1.1.4 INT\_SIZE

```
#define INT_SIZE 4
```

### 2.1.1.5 RCVBUFSIZE

```
#define RCVBUFSIZE 10240
```

### 2.1.1.6 SHMOBJ\_PATH

```
#define SHMOBJ_PATH "/ipc_obj"
```

## 2.1.2 Function Documentation

### 2.1.2.1 getResponse()

```
void getResponse (
    int sckt,
    char * filename,
    char * fileType )
```

Response for a GET request



## Parameters

<i>sckt</i>	The server socket connected to this client.
<i>filename</i>	The name of the file that should be sent to the client .
<i>fileType</i>	The type of the file to be sent to client .

## 2.1.2.2 interruptHandler()

```
void interruptHandler (  
    int signal )
```

- Checks client activity Handles interrupt signal

## Parameters

<i>signal</i>	signal type
---------------	-------------

## 2.1.2.3 main()

```
int main (  
    int argc,  
    char * argv[] )
```

TCP client handling function

## 2.1.2.4 postResponse()

```
void postResponse (  
    int sckt,  
    char * filename,  
    int file_size )
```

Response for a POST request

## Parameters

<i>sckt</i>	The server socket connected to this client.
<i>filename</i>	The name of the file sent by the client.
<i>file_size</i>	The size of the file to be sent by the client .

### 2.1.2.5 print()

```
void print (
    char * str )
```

Logging function

#### Parameters

<i>str</i>	String that needs to be outputed on the terminal for logging reasons.
------------	---

### 2.1.2.6 responseForClient()

```
void responseForClient (
    int sckt )
```

Top level response function

#### Parameters

<i>sckt</i>	The server socket number connected to this client.
-------------	--

### 2.1.2.7 sendBytes()

```
void sendBytes (
    int sckt,
    int fileSize,
    int filed )
```

Function that sends files as a ByteStream

#### Parameters

<i>sckt</i>	The server socket connected to this client.
<i>fileSize</i>	The size of the file to be sent to client.
<i>filed</i>	The file descriptor from which data is taken to be sent to client.

### 2.1.2.8 sendMessageToClient()

```
void sendMessageToClient (
    int sckt,
```

```
int code,  
int file_size )
```

Sending HTTP reply codes

#### Parameters

<i>sckt</i>	The server socket connected to this client.
<i>code</i>	The HTTP code to be sent to the client.
<i>file_size</i>	The size of the file to be sent/received by the client.



# Index

CLIENTS\_IN\_QUEUE

HTTPServer.c, [4](#)

CONNECTION\_TIME\_OUT\_Sec

HTTPServer.c, [4](#)

CONNECTION\_TIME\_OUT\_USec

HTTPServer.c, [4](#)

getResponse

HTTPServer.c, [4](#)

HTTPServer.c, [3](#)

CLIENTS\_IN\_QUEUE, [4](#)

CONNECTION\_TIME\_OUT\_Sec, [4](#)

CONNECTION\_TIME\_OUT\_USec, [4](#)

getResponse, [4](#)

INT\_SIZE, [4](#)

interruptHandler, [5](#)

main, [5](#)

postResponse, [5](#)

print, [5](#)

RCVBUFSIZE, [4](#)

responseForClient, [6](#)

SHMOBJ\_PATH, [4](#)

sendBytes, [6](#)

sendMessageToClient, [6](#)

INT\_SIZE

HTTPServer.c, [4](#)

interruptHandler

HTTPServer.c, [5](#)

main

HTTPServer.c, [5](#)

postResponse

HTTPServer.c, [5](#)

print

HTTPServer.c, [5](#)

RCVBUFSIZE

HTTPServer.c, [4](#)

responseForClient

HTTPServer.c, [6](#)

SHMOBJ\_PATH

HTTPServer.c, [4](#)

sendBytes

HTTPServer.c, [6](#)

sendMessageToClient

HTTPServer.c, [6](#)