HTTP server

1.2

Generated by Doxygen 1.8.13

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

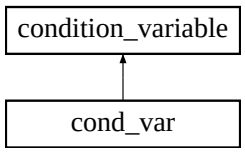
Class Documentation

3.1 cond_var Class Reference

condition variables are used for communication between threads

`#include <Servers.h>`

Inheritance diagram for cond_var:



Public Member Functions

- cond_var ()
  standard constructor
- ∼cond_var ()
  standard destructor, no extra memory is used
- void set_data (boost::asio::io_context &iocont, server ∗serv, session ∗s)
  initializes the values of this class' object's members
- boost::asio::io_context ∗ get_context ()
  returns the current io_context
- server ∗ get_server ()
  returns the pointer to current server
- session ∗ get_session ()
  returns the pointer to current session

3.1.1 Detailed Description

condition variables are used for communication between threads

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

- Servers.h

3.2 page Class Reference

page class contains array of paragraphs and function to work with them

`#include <Servers.h>`

Public Member Functions

- page ()
  default constructor that puts 0's to member fields
- page (char *pname)
- ~page ()
  destructor that calls del_page
- char * get_page_name ()
  returns current page name
- bool is_page (char *name)
  compares current page name with name
- int open_page (char *pname)
- int delete_page ()
  deletes current page, returns 0 if OK, 1 if not
- int del_paragraph (int num)
- int add_to_par (int num, char *text)
- int insert_paragraph_at_end (char *text)
  adds new paragraph at the end of the array, returns length of text if OK, -1 if not
- int write_page ()
  writes changes of the page on the disk, returns 0 if OK, 1 if not

Static Public Member Functions

- static int delete_page (char *name)
  deletes page from the drive by its name

Protected Member Functions

- int open_page ()
  hiden function to open page by the name in pagename
- int del_page ()
  hiden function to destroy all allocated memory if any

Protected Attributes

- std::array< paragraph, 100 > **par_arr**
- int **numpar**
- char * **pagename**

3.2.1 Detailed Description

page class contains array of paragraphs and function to work with them

3.2.2 Constructor & Destructor Documentation

3.2.2.1 page()

```
page::page (
            char * pname )  [inline]
```

constructor pname — name of the page to open and process stores pname in pagename, also opens this page

3.2.3 Member Function Documentation

3.2.3.1 add_to_par()

```
int page::add_to_par (
            int num,
            char * text )
```

adds text to paragraph num — the number of the paragraph to which we add text text — text to add to paragraph, returns length of new paragraph if OK, -1 if not

3.2.3.2 del_paragraph()

```
int page::del_paragraph (
            int num )
```

deletes current paragraph by its number num — the number of paragraph to delete, returns 0 if OK, 1 if not

3.2.3.3 open_page()

```
int page::open_page (
            char * pname )
```

opens page pname — name of the page to open

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

- Servers.h
- main.cpp

3.3 paragraph Class Reference

paragraph class to contain paragraphs of the text

```
#include <Servers.h>
```

Public Member Functions

- paragraph ()
  default constructor that puts nullptr into str
- paragraph (char ∗s)
- ∼paragraph ()
  default destructor
- int fill_par (char ∗val)
- int del_par ()
  deletes paragraph and frees memory. returns 0 if OK, 1 if is empty
- int add_text_to_par (char ∗text)
  adds text to the end of the paragraph
- int get_length ()
  gets length in bytes of the current paragraph
- char ∗ get_par ()
  gets current paragraph entity

Protected Attributes

- char ∗ str
  paragraph's text storage
- int len
  length of the text in paragraph

3.3.1 Detailed Description

paragraph class to contain paragraphs of the text

Used in page class.

3.3.2 Constructor & Destructor Documentation

3.3.2.1 paragraph()

```
paragraph::paragraph (
            char * s )  [inline]
```

constructor that stores the paragraph. copies incoming text from s and stores it into str.

3.3.3 Member Function Documentation

3.3.3.1 fill_par()

```
int paragraph::fill_par (
            char * val )
```

returns the length of loaded paragraph after removing old text and replacing it by new incoming text from val

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

- Servers.h
- main.cpp

3.4 server Class Reference

HTTP server's main class.

```
#include <Servers.h>
```

Public Member Functions

- server (boost::asio::io_context &io_context, short port, int num_threads)
- ~server ()
  standard destructor, runs free_resources
- tcp::acceptor & get_acceptor ()
  returns current acceptor
- void handle_accept (session *new_session, const boost::system::error_code &error, int num)

3.4.1 Detailed Description

HTTP server's main class.

3.4.2 Constructor & Destructor Documentation

3.4.2.1 server()

```
server::server (
            boost::asio::io_context & io_context,
            short port,
            int num_threads )  [inline]
```

standard constructor, runs initializer accepts opened io_context port — number, default is 80 the number of threads to run

3.4.3 Member Function Documentation

3.4.3.1 handle_accept()

```
void server::handle_accept (
            session * new_session,
            const boost::system::error_code & error,
            int num )
```

function that is called in answer to the event of accepting new incoming connection gets link to new_session object, error code and the number of the thread

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

- Servers.h
- main.cpp

3.5 server_manager Class Reference

Public Member Functions

- int **cmline_parser** (int argc, char ∗argv[ ])
- **server_manager** (int argc, char ∗argv[ ])
- int **run_server_test** ()

Public Attributes

- variables_map **vm**
- int **v** =0

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

- server_manager.cpp

3.6 session Class Reference

operates with requests and sends the respond

```
#include <Servers.h>
```

Public Member Functions

- session (boost::asio::io_context &io_context, int thr_num)
- tcp::socket & socket ()

  returns current opened socket
- void start ()

  function that initializes waiting for requests and all variables
- int get_thread_num ()

  returns current thread's number

3.6.1 Detailed Description

operates with requests and sends the respond

3.6.2 Constructor & Destructor Documentation

3.6.2.1 session()

```
session::session (
            boost::asio::io_context & io_context,
            int thr_num )  [inline]
```

standard constructor io_context — context of opened connection thr_num — this thread's number to store

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

- Servers.h
- main.cpp

Index