NetSocket++

0.1

Generated by Doxygen 1.8.3.1

Wed Feb 27 2013 00:04:36

Contents

1 Namespace Index **1**

1.1 Namespace List 1

2 Hierarchical Index **3**

2.1 Class Hierarchy 3

3 Class Index **5**

3.1 Class List 5

4 File Index **7**

4.1 File List 7

5 Namespace Documentation **9**

5.1 NetSocketPP Namespace Reference 9

5.1.1 Detailed Description 9

5.1.2 Function Documentation 10

5.1.2.1 CStrToString 10

6 Class Documentation **11**

6.1 NetSocketPP::ClientSocket Class Reference 11

6.1.1 Detailed Description 11

6.1.2 Constructor & Destructor Documentation 12

6.1.2.1 ClientSocket 12

6.1.3 Member Function Documentation 12

6.1.3.1 get 12

6.1.3.2 recv 12

6.1.3.3 send 12

6.2 NetSocketPP::HTTPClientSocket Class Reference 13

6.2.1 Detailed Description 13

6.2.2 Constructor & Destructor Documentation 13

6.2.2.1 HTTPClientSocket 13

6.2.3 Member Function Documentation 13

6.2.3.1 getReply 13

6.2.3.2 getRequest . . . 14
6.3 NetSocketPP::HTTPReply Class Reference . . . 14
6.3.1 Detailed Description . . . 15
6.3.2 Constructor & Destructor Documentation . . . 15
6.3.2.1 HTTPReply . . . 15
6.3.3 Member Function Documentation . . . 15
6.3.3.1 addToContent . . . 15
6.3.3.2 getConnection . . . 15
6.3.3.3 getContent . . . 15
6.3.3.4 getContentLength . . . 15
6.3.3.5 getContentType . . . 15
6.3.3.6 getProtocol . . . 16
6.3.3.7 getRaw . . . 16
6.3.3.8 getResponse . . . 16
6.3.3.9 getServer . . . 16
6.3.3.10 getTimestamp . . . 16
6.4 NetSocketPP::NetSocket Class Reference . . . 16
6.4.1 Detailed Description . . . 18
6.4.2 Constructor & Destructor Documentation . . . 18
6.4.2.1 NetSocket . . . 18
6.4.3 Member Function Documentation . . . 18
6.4.3.1 getDesc . . . 18
6.4.3.2 getIP . . . 18
6.5 NetSocketPP::NetworkException Class Reference . . . 18
6.5.1 Detailed Description . . . 19
6.5.2 Constructor & Destructor Documentation . . . 19
6.5.2.1 NetworkException . . . 19
6.5.3 Member Function Documentation . . . 19
6.5.3.1 what . . . 19
6.6 NetSocketPP::ServerFunctionArgs Class Reference . . . 19
6.6.1 Detailed Description . . . 20
6.6.2 Constructor & Destructor Documentation . . . 20
6.6.2.1 ServerFunctionArgs . . . 20
6.6.3 Member Function Documentation . . . 20
6.6.3.1 addArgument . . . 20
6.6.3.2 getArgument . . . 20
6.6.3.3 operator[] . . . 20
6.7 NetSocketPP::ServerSocket Class Reference . . . 21
6.7.1 Detailed Description . . . 21
6.7.2 Constructor & Destructor Documentation . . . 21

6.7.2.1 ServerSocket . . . 21
6.7.3 Member Function Documentation . . . 22
6.7.3.1 get . . . 22
6.7.3.2 recv . . . 22
6.7.3.3 send . . . 22
6.7.3.4 startServer . . . 22
6.8 NetSocketPP::SocketException Class Reference . . . 23
6.8.1 Detailed Description . . . 23
6.8.2 Constructor & Destructor Documentation . . . 23
6.8.2.1 SocketException . . . 23
6.8.3 Member Function Documentation . . . 23
6.8.3.1 what . . . 23

7 File Documentation **25**
7.1 ClientSocket.h File Reference . . . 25
7.1.1 Detailed Description . . . 25
7.2 HTTPClientSocket.h File Reference . . . 25
7.2.1 Detailed Description . . . 26
7.3 NetSocket.h File Reference . . . 26
7.3.1 Detailed Description . . . 27
7.4 NetSocketPP.h File Reference . . . 27
7.4.1 Detailed Description . . . 27
7.5 NetworkException.h File Reference . . . 28
7.5.1 Detailed Description . . . 28
7.6 ServerSocket.h File Reference . . . 28
7.6.1 Detailed Description . . . 29
7.6.2 Function Documentation . . . 29
7.6.2.1 sigchld_handler . . . 29
7.7 SocketException.h File Reference . . . 29
7.7.1 Detailed Description . . . 29

Index **30**

Chapter 1

Namespace Index

1.1 Namespace List

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

NetSocketPP
A namespace for all library names 9

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

exception
- NetSocketPP::NetworkException . . . 18
- NetSocketPP::SocketException . . . 23

NetSocketPP::HTTPReply . . . 14
NetSocketPP::NetSocket . . . 16
- NetSocketPP::ClientSocket . . . 11
  - NetSocketPP::HTTPClientSocket . . . 13
- NetSocketPP::ServerSocket . . . 21

NetSocketPP::ServerFunctionArgs . . . 19

Chapter 3

Class Index

3.1 Class List

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

NetSocketPP::ClientSocket
: An implementation of a client socket. Inherits from NetSocket 11

NetSocketPP::HTTPClientSocket
: A class representing HTTP client socket 13

NetSocketPP::HTTPReply
: A class representing HTTP Reply 14

NetSocketPP::NetSocket
: A class, that represents network connection - socket 16

NetSocketPP::NetworkException
: A class representing an exception with network 18

NetSocketPP::ServerFunctionArgs
: A class for storing server function arguments 19

NetSocketPP::ServerSocket
: An implementation of the server socket 21

NetSocketPP::SocketException
: A class representing an exception with socket classes 23

Chapter 4

File Index

4.1 File List

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

ClientSocket.h
An implementation of a client socket 25
HTTPClientSocket.h
An implementation of HTTP Client Socket 25
NetSocket.h
A library designed to simplify the use of UNIX Network Sockets in the means of OOP 26
NetSocketPP.h
A common header for NetSocket++ library 27
NetworkException.h
An implementation of network exception 28
ServerSocket.h
An implementation of a server socket 28
SocketException.h
An implementation of socket exception 29

Chapter 5

Namespace Documentation

5.1 NetSocketPP Namespace Reference

A namespace for all library names.

Classes

- class ClientSocket

  An implementation of a client socket. Inherits from NetSocket.
- class HTTPReply

  A class representing HTTP Reply.
- class HTTPClientSocket

  A class representing HTTP client socket.
- class NetSocket

  A class, that represents network connection - socket.
- class NetworkException

  A class representing an exception with network.
- class ServerFunctionArgs

  A class for storing server function arguments.
- class ServerSocket

  An implementation of the server socket.
- class SocketException

  A class representing an exception with socket classes.

Functions

- std::string CStrToString (char *cstr)

  A function, that converts table of chars (a C-style string) into std::string.

5.1.1 Detailed Description

A namespace for all library names.

5.1.2 Function Documentation

5.1.2.1 NetSocketPP::CStrToString (char * *cstr*) [inline]

A function, that converts table of chars (a C-style string) into std::string.

Parameters

cstr	A C-style string to be converted.

Returns

A std::string with the content of the input.

Chapter 6

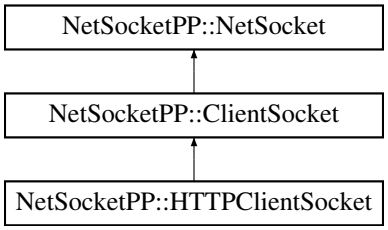
Class Documentation

6.1 NetSocketPP::ClientSocket Class Reference

An implementation of a client socket. Inherits from NetSocket.

`#include <ClientSocket.h>`

Inheritance diagram for NetSocketPP::ClientSocket:



Public Member Functions

- ClientSocket (std::string host, std::string service, std::string protocol)

  A constructor with parameters, that creates and connects the socket.
- int send (std::string msg, int flags)

  A function, that sends data through the socket.
- int recv (int flags)

  A function, that receives data through the socket.
- std::string get ()

  A function returning recently recv-d data.

Protected Attributes

- char buf [100000]

  A large buffer for data.

Additional Inherited Members

6.1.1 Detailed Description

An implementation of a client socket. Inherits from NetSocket.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 ClientSocket::ClientSocket (std::string *host,* std::string *service,* std::string *protocol*)

A constructor with parameters, that creates and connects the socket.

Parameters

host	A hostname or IP address of socket destination.
service	A port or service identifier, where socket is to be opened.
protocol	A protocol of the socket, TCP or UDP.

6.1.3 Member Function Documentation

6.1.3.1 std::string ClientSocket::get ( )

A function returning recently recv-d data.

Returns

Received data as std::string.

6.1.3.2 int ClientSocket::recv (int *flags*)

A function, that receives data through the socket.

Parameters

flags	Socket flags, default 0.

Returns

Number of bytes received.

6.1.3.3 int ClientSocket::send (std::string *msg,* int *flags* = 0)

A function, that sends data through the socket.

Parameters

msg	A message to send.
flags	Socket flags, default 0.

Returns

Number of bytes sent.

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

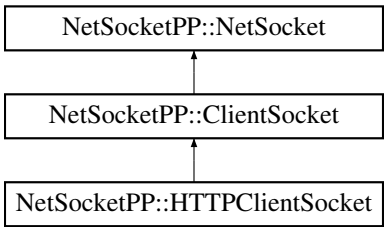
- ClientSocket.h
- ClientSocket.cpp

6.2 NetSocketPP::HTTPClientSocket Class Reference

A class representing HTTP client socket.

`#include <HTTPClientSocket.h>`

Inheritance diagram for NetSocketPP::HTTPClientSocket:



Public Member Functions

- HTTPClientSocket (std::string host, std::string service, std::string docRequest)

  A constructor with parameters.
- HTTPReply getReply ()

  A function returning a HTTPReply.
- std::string getRequest ()

  A function returning the request used in the socket.

Additional Inherited Members

6.2.1 Detailed Description

A class representing HTTP client socket.

6.2.2 Constructor & Destructor Documentation

6.2.2.1 HTTPClientSocket::HTTPClientSocket (std::string *host* = `NULL`, std::string *service* = `"http"`, std::string *docRequest* = `"/"`)

A constructor with parameters.

Parameters

host	Hostname or IP of socket destination, defaults to NULL.
service	Service port or identifier, defaults to HTTP.
docRequest	A document to request from the server, defaults to root/index (/).

6.2.3 Member Function Documentation

6.2.3.1 HTTPReply HTTPClientSocket::getReply ( )

A function returning a HTTPReply.

Returns

HTTPReply object containing received data.

6.2.3.2 std::string HTTPClientSocket::getRequest ( )

A function returning the request used in the socket.

Returns

The HTTP request used to obtain data.

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

- HTTPClientSocket.h
- HTTPClientSocket.cpp

6.3 NetSocketPP::HTTPReply Class Reference

A class representing HTTP Reply.

`#include <HTTPClientSocket.h>`

Public Member Functions

- HTTPReply ()
  A constructor.
- HTTPReply (std::string raw)
  A constructor with parameter.
- ~HTTPReply ()
  A destructor.
- void parse ()
  HTTP reply parser.
- void addToContent (std::string cp)
  A function, that adds more parts of the content to the reply if necessary.
- std::string getRaw ()
  A function returning raw HTTP reply.
- std::string getProtocol ()
  A function returning HTTP protocol information.
- std::string getResponse ()
  A function returning HTTP response message.
- std::string getTimestamp ()
  A function returning timestamp.
- std::string getServer ()
  A function returning server information.
- unsigned int getContentLength ()
  A function returning length of content.
- std::string getConnection ()
  A function returning connection status.
- std::string getContentType ()
  A function returning type of content.
- std::string getContent ()
  A function returning received content.

6.3.1 Detailed Description

A class representing HTTP Reply.

6.3.2 Constructor & Destructor Documentation

6.3.2.1 HTTPReply::HTTPReply (std::string *raw*)

A constructor with parameter.

Parameters

raw	Raw reply from recv.

6.3.3 Member Function Documentation

6.3.3.1 void HTTPReply::addToContent (std::string *cp*)

A function, that adds more parts of the content to the reply if necessary.

Parameters

cp	Part of the content to be added.

6.3.3.2 std::string HTTPReply::getConnection ( )

A function returning connection status.

Returns

Connection status.

6.3.3.3 std::string HTTPReply::getContent ( )

A function returning received content.

Returns

Received content.

6.3.3.4 unsigned int HTTPReply::getContentLength ( )

A function returning length of content.

Returns

Length of content.

6.3.3.5 std::string HTTPReply::getContentType ( )

A function returning type of content.

Returns

Type of content.

6.3.3.6 std::string HTTPReply::getProtocol ( )

A function returning HTTP protocol information.

Returns

HTTP protocol information.

6.3.3.7 std::string HTTPReply::getRaw ( )

A function returning raw HTTP reply.

Returns

Raw HTTP reply.

6.3.3.8 std::string HTTPReply::getResponse ( )

A function returning HTTP response message.

Returns

HTTP response message.

6.3.3.9 std::string HTTPReply::getServer ( )

A function returning server information.

Returns

Server information.

6.3.3.10 std::string HTTPReply::getTimestamp ( )

A function returning timestamp.

Returns

Timestamp.

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

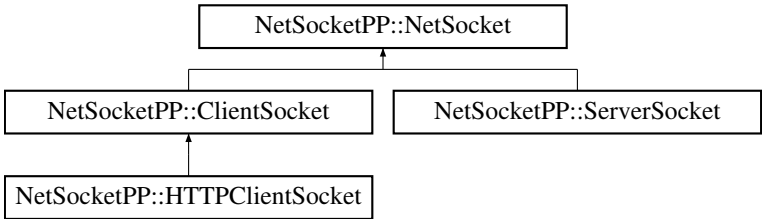
- HTTPClientSocket.h
- HTTPClientSocket.cpp

6.4 NetSocketPP::NetSocket Class Reference

A class, that represents network connection - socket.

`#include <NetSocket.h>`

Inheritance diagram for NetSocketPP::NetSocket:



Public Member Functions

- NetSocket (std::string host, std::string service, std::string protocol)

  A constructor with parameters, that creates a socket.
- std::string getIP ()

  A function that returns IP of a host.
- int getDesc ()

  A function that returns socket descriptor.
- ~NetSocket ()

  A destructor, that frees the memory.

Protected Member Functions

- void * get_in_addr (sockaddr *sa)

  Needed for implementation purposes.

Protected Attributes

- int _descriptor

  Socket descriptor.
- int _yes

  Needed for implementation purposes.
- int _status

  Needed for implementation purposes.
- char _caddr [INET6_ADDRSTRLEN]

  A structure that stores IP address.
- addrinfo _hints

  Needed for implementation purposes.
- addrinfo * _servinfo

  Needed for implementation purposes.
- sockaddr_storage _their_addr

  Needed for implementation purposes.
- socklen_t _addr_size

  Needed for implementation purposes.
- std::string _host

  A host to which a socket is connecting to/on which a server socket is opened.
- std::string _service

  A port or a string identyfing service that socket is connecting to/which server is being opened.
- std::string _protocol

  A protocol of the socket: TCP/UDP.

6.4.1 Detailed Description

A class, that represents network connection - socket.

6.4.2 Constructor & Destructor Documentation

6.4.2.1 NetSocket::NetSocket (std::string *host*, std::string *service*, std::string *protocol*)

A constructor with parameters, that creates a socket.

Parameters

host	A hostname or IP address of socket destination.
service	A port or service identifier, where socket is to be opened.
protocol	A protocol of the socket, TCP or UDP.

6.4.3 Member Function Documentation

6.4.3.1 int NetSocket::getDesc ( )

A function that returns socket descriptor.

Returns

A socket descriptor.

6.4.3.2 std::string NetSocket::getIP ( )

A function that returns IP of a host.

Returns

IP address of a host as std::string.

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

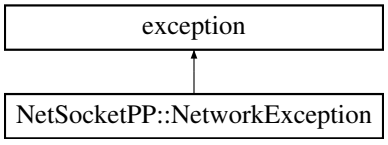
- NetSocket.h
- NetSocket.cpp

6.5 NetSocketPP::NetworkException Class Reference

A class representing an exception with network.

`#include <NetworkException.h>`

Inheritance diagram for NetSocketPP::NetworkException:



Public Member Functions

- NetworkException (std::string cmd, std::string msg)

  A constructor with parameters.
- ~NetworkException () throw ()

  A destructor, as needed by std::exception.
- const char * what () const throw ()

  A function, that returns error message, as needed by std::exception.

6.5.1 Detailed Description

A class representing an exception with network.

6.5.2 Constructor & Destructor Documentation

6.5.2.1 NetworkException::NetworkException (std::string *cmd*, std::string *msg*)

A constructor with parameters.

Parameters

cmd	A command, where exception occured.
msg	What has happened.

6.5.3 Member Function Documentation

6.5.3.1 const char * NetworkException::what ( ) const throw ()

A function, that returns error message, as needed by std::exception.

Returns

Error message.

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

- NetworkException.h
- NetworkException.cpp

6.6 NetSocketPP::ServerFunctionArgs Class Reference

A class for storing server function arguments.

`#include <ServerSocket.h>`

Public Member Functions

- ServerFunctionArgs ()

  A constructor.
- ServerFunctionArgs (ServerFunctionArgs &sfa)

  A copy constructor.
- ~ServerFunctionArgs ()

A destructor.

- void addArgument (std::string arg)

  Function adding an argument to the list.

- std::string getArgument (unsigned int idx)

  Function returning the argument of given index number.

- std::string operator[] (unsigned int idx)

  Operator[] returning the argument of given index number.

6.6.1 Detailed Description

A class for storing server function arguments.

6.6.2 Constructor & Destructor Documentation

6.6.2.1 ServerFunctionArgs::ServerFunctionArgs (ServerFunctionArgs & *sfa*)

A copy constructor.

Parameters

sfa	An object to be copied.

6.6.3 Member Function Documentation

6.6.3.1 void ServerFunctionArgs::addArgument (std::string *arg*)

Function adding an argument to the list.

Parameters

arg	An argument to be added, of type std::string.

6.6.3.2 std::string ServerFunctionArgs::getArgument (unsigned int *idx*)

Function returning the argument of given index number.

Parameters

idx	Index of the argument.

Returns

The argument.

6.6.3.3 std::string ServerFunctionArgs::operator[] (unsigned int *idx*)

Operator[] returning the argument of given index number.

Parameters

idx	Index of the argument.

Returns

The argument.

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

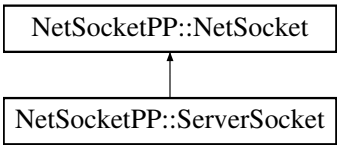
- ServerSocket.h
- ServerSocket.cpp

6.7 NetSocketPP::ServerSocket Class Reference

An implementation of the server socket.

`#include <ServerSocket.h>`

Inheritance diagram for NetSocketPP::ServerSocket:



Public Member Functions

- ServerSocket (std::string host, std::string service, std::string protocol)

  A constructor with parameters.
- ~ServerSocket ()

  A destructor.
- void startServer (ServerFunctionArgs &functionOutput, ServerFunctionArgs &(*serverMain)(ServerFunctionArgs, ServerSocket *), ServerFunctionArgs functionInput, bool infinite, unsigned int iternum, int connectionLimit)

  A function that starts TCP server.
- int send (std::string msg, int flags=0)

  A function that sends data through the socket.
- int recv (int flags=0)

  A function that receives data through the socket.
- std::string get ()

  A function returning received data.

Additional Inherited Members

6.7.1 Detailed Description

An implementation of the server socket.

6.7.2 Constructor & Destructor Documentation

6.7.2.1 ServerSocket::ServerSocket (std::string *host,* std::string *service,* std::string *protocol*)

A constructor with parameters.

Parameters

host	A hostname or IP adress of socket destination, defaults to NULL.
service	Port or service that socket should be connected with.
protocol	Socket protocol, TCP or UDP.

6.7.3 Member Function Documentation

6.7.3.1 std::string ServerSocket::get ( )

A function returning received data.

Returns

Received data as string.

6.7.3.2 int ServerSocket::recv (int *flags* = 0)

A function that receives data through the socket.

Parameters

flags	Receive flags, defaulting to 0.

Returns

Number of bytes received.

6.7.3.3 int ServerSocket::send (std::string *msg*, int *flags* = 0)

A function that sends data through the socket.

Parameters

msg	A message/data to send, of type std::string.
flags	Send flags, defaulting to 0.

Returns

Number of bytes sent.

6.7.3.4 void ServerSocket::startServer (ServerFunctionArgs & *functionOutput*, ServerFunctionArgs &(∗)(ServerFunctionArgs, ServerSocket ∗) *serverMain*, ServerFunctionArgs *functionInput*, bool *infinite*, unsigned int *iternum*, int *connectionLimit*)

A function that starts TCP server.

Parameters

functionOutput	A ServerFunctionArgs object that will store server function result.
serverMain	An user-defined function, that returns ServerFunctionArgs object - results of the server function with arguments: ServerFunctionArgs object - arguments to the server function and pointer to ServerSocket object - for passing socket information in that order.
functionInput	A ServerFunctionArgs object with server function arguments.
infinite	Determines if server loop should be infinite.

iternum	Number of accept() iterations for non-infinite loops.
connectionLimit	Maximum number of accepted connections.

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

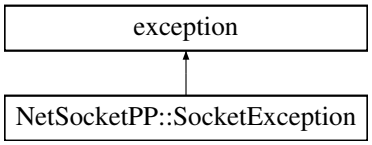
- ServerSocket.h
- ServerSocket.cpp

6.8 NetSocketPP::SocketException Class Reference

A class representing an exception with socket classes.

`#include <SocketException.h>`

Inheritance diagram for NetSocketPP::SocketException:



Public Member Functions

- SocketException (std::string msg)

  A constructor with parameters.
- ~SocketException () throw ()

  A destructor, as needed by std::exception.
- const char * what () const throw ()

  A function, that returns error message, as needed by std::exception.

6.8.1 Detailed Description

A class representing an exception with socket classes.

6.8.2 Constructor & Destructor Documentation

6.8.2.1 SocketException::SocketException (std::string *msg*)

A constructor with parameters.

Parameters

msg	What has happened.

6.8.3 Member Function Documentation

6.8.3.1 const char * SocketException::what ( ) const throw ()

A function, that returns error message, as needed by std::exception.

Returns

Error message.

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

- SocketException.h
- SocketException.cpp

Chapter 7

File Documentation

7.1 ClientSocket.h File Reference

An implementation of a client socket.

```
#include "NetSocket.h"
#include "NetworkException.h"
```

Classes

- class NetSocketPP::ClientSocket

  An implementation of a client socket. Inherits from NetSocket.

Namespaces

- namespace NetSocketPP

  A namespace for all library names.

7.1.1 Detailed Description

An implementation of a client socket.

Author

Phitherek_

Date

2012

Version

0.1

7.2 HTTPClientSocket.h File Reference

An implementation of HTTP Client Socket.

```
#include "ClientSocket.h"
#include "SocketException.h"
```

Classes

- class NetSocketPP::HTTPReply

  A class representing HTTP Reply.
- class NetSocketPP::HTTPClientSocket

  A class representing HTTP client socket.

Namespaces

- namespace NetSocketPP

  A namespace for all library names.

7.2.1 Detailed Description

An implementation of HTTP Client Socket.

Author

Phitherek_

Date

2012

Version

0.1

7.3 NetSocket.h File Reference

A library designed to simplify the use of UNIX Network Sockets in the means of OOP.

```
#include <sys/socket.h>
#include <sys/types.h>
#include <arpa/inet.h>
#include <netdb.h>
#include <unistd.h>
#include <sys/wait.h>
#include <signal.h>
#include <string>
#include <cerrno>
#include <cstring>
```

Classes

- class NetSocketPP::NetSocket

  A class, that represents network connection - socket.

Namespaces

- namespace NetSocketPP

  A namespace for all library names.

Functions

- std::string NetSocketPP::CStrToString (char *cstr)

  A function, that converts table of chars (a C-style string) into std::string.

7.3.1 Detailed Description

A library designed to simplify the use of UNIX Network Sockets in the means of OOP.

Author

Phitherek_

Date

2012

Version

0.1

7.4 NetSocketPP.h File Reference

A common header for NetSocket++ library.

```
#include "NetSocket.h"
#include "SocketException.h"
#include "NetworkException.h"
#include "ClientSocket.h"
#include "ServerSocket.h"
#include "HTTPClientSocket.h"
```

7.4.1 Detailed Description

A common header for NetSocket++ library.

Author

Phitherek_

Date

2013

Version

0.1

7.5 NetworkException.h File Reference

An implementation of network exception.

```
#include <exception>
#include <string>
```

Classes

- class NetSocketPP::NetworkException
  A class representing an exception with network.

Namespaces

- namespace NetSocketPP
  A namespace for all library names.

7.5.1 Detailed Description

An implementation of network exception.

Author

Phitherek_

Date

2012

Version

0.1

7.6 ServerSocket.h File Reference

An implementation of a server socket.

```
#include "NetSocket.h"
#include "NetworkException.h"
#include "SocketException.h"
```

Classes

- class NetSocketPP::ServerFunctionArgs
  A class for storing server function arguments.
- class NetSocketPP::ServerSocket
  An implementation of the server socket.

Namespaces

- namespace NetSocketPP
  A namespace for all library names.

Functions

- void sigchld_handler (int s)

  Signal handler, needed for implementation purposes.

7.6.1 Detailed Description

An implementation of a server socket.

Author

Phitherek_

Date

2013

Version

0.1

7.6.2 Function Documentation

7.6.2.1 sigchld_handler (int *s*) [inline]

Signal handler, needed for implementation purposes.

Parameters

s	Needed for implementation purposes

7.7 SocketException.h File Reference

An implementation of socket exception.

```
#include <exception>
#include <string>
```

Classes

- class NetSocketPP::SocketException

  A class representing an exception with socket classes.

Namespaces

- namespace NetSocketPP

  A namespace for all library names.

7.7.1 Detailed Description

An implementation of socket exception.

Author

Phitherek_

Date

2012

Version

0.1

Index

addArgument
    NetSocketPP::ServerFunctionArgs, 20
addToContent
    NetSocketPP::HTTPReply, 15

CStrToString
    NetSocketPP, 10
ClientSocket
    NetSocketPP::ClientSocket, 12
ClientSocket.h, 25

get
    NetSocketPP::ClientSocket, 12
    NetSocketPP::ServerSocket, 22
getArgument
    NetSocketPP::ServerFunctionArgs, 20
getConnection
    NetSocketPP::HTTPReply, 15
getContent
    NetSocketPP::HTTPReply, 15
getContentLength
    NetSocketPP::HTTPReply, 15
getContentType
    NetSocketPP::HTTPReply, 15
getDesc
    NetSocketPP::NetSocket, 18
getIP
    NetSocketPP::NetSocket, 18
getProtocol
    NetSocketPP::HTTPReply, 15
getRaw
    NetSocketPP::HTTPReply, 16
getReply
    NetSocketPP::HTTPClientSocket, 13
getRequest
    NetSocketPP::HTTPClientSocket, 13
getResponse
    NetSocketPP::HTTPReply, 16
getServer
    NetSocketPP::HTTPReply, 16
getTimestamp
    NetSocketPP::HTTPReply, 16

HTTPClientSocket
    NetSocketPP::HTTPClientSocket, 13
HTTPClientSocket.h, 25
HTTPReply
    NetSocketPP::HTTPReply, 15

NetSocket
    NetSocketPP::NetSocket, 18
NetSocket.h, 26
NetSocketPP, 9
    CStrToString, 10
NetSocketPP.h, 27
NetSocketPP::ClientSocket, 11
    ClientSocket, 12
    get, 12
    recv, 12
    send, 12
NetSocketPP::HTTPClientSocket, 13
    getReply, 13
    getRequest, 13
    HTTPClientSocket, 13
NetSocketPP::HTTPReply, 14
    addToContent, 15
    getConnection, 15
    getContent, 15
    getContentLength, 15
    getContentType, 15
    getProtocol, 15
    getRaw, 16
    getResponse, 16
    getServer, 16
    getTimestamp, 16
    HTTPReply, 15
NetSocketPP::NetSocket, 16
    getDesc, 18
    getIP, 18
    NetSocket, 18
NetSocketPP::NetworkException, 18
    NetworkException, 19
    what, 19
NetSocketPP::ServerFunctionArgs, 19
    addArgument, 20
    getArgument, 20
    ServerFunctionArgs, 20
NetSocketPP::ServerSocket, 21
    get, 22
    recv, 22
    send, 22
    ServerSocket, 21
    startServer, 22
NetSocketPP::SocketException, 23
    SocketException, 23
    what, 23
NetworkException
    NetSocketPP::NetworkException, 19
NetworkException.h, 28

recv
NetSocketPP::ClientSocket, 12
NetSocketPP::ServerSocket, 22

send
NetSocketPP::ClientSocket, 12
NetSocketPP::ServerSocket, 22
ServerFunctionArgs
NetSocketPP::ServerFunctionArgs, 20
ServerSocket
NetSocketPP::ServerSocket, 21
ServerSocket.h, 28
sigchld_handler, 29
sigchld_handler
ServerSocket.h, 29
SocketException
NetSocketPP::SocketException, 23
SocketException.h, 29
startServer
NetSocketPP::ServerSocket, 22

what
NetSocketPP::NetworkException, 19
NetSocketPP::SocketException, 23