FreeCoAP

Generated by Doxygen 1.8.10

Fri Aug 21 2015 21:29:13

Contents

1 README **1**

2 Class Index **3**

2.1 Class List 3

3 File Index **5**

3.1 File List 5

4 Class Documentation **7**

4.1 coap_client_t Struct Reference 7

4.1.1 Detailed Description 7

4.1.2 Member Data Documentation 7

4.1.2.1 num_retrans 7

4.1.2.2 sd 7

4.1.2.3 server_addr 7

4.1.2.4 server_sin 7

4.1.2.5 server_sin_len 8

4.1.2.6 timeout 8

4.1.2.7 timer_fd 8

4.2 coap_msg_op Struct Reference 8

4.2.1 Detailed Description 8

4.2.2 Member Data Documentation 8

4.2.2.1 len 8

4.2.2.2 next 8

4.2.2.3 num 8

4.2.2.4 val 9

4.3 coap_msg_op_list_t Struct Reference 9

4.3.1 Detailed Description 9

4.3.2 Member Data Documentation 9

4.3.2.1 first 9

4.3.2.2 last 9

4.4 coap_msg_t Struct Reference 9

4.4.1 Detailed Description 10
4.4.2 Member Data Documentation 10
4.4.2.1 code_class 10
4.4.2.2 code_detail 10
4.4.2.3 msg_id 10
4.4.2.4 op_list 10
4.4.2.5 payload 10
4.4.2.6 payload_len 10
4.4.2.7 token 10
4.4.2.8 token_len 10
4.4.2.9 type 10
4.4.2.10 ver 11
4.5 coap_server Struct Reference 11
4.5.1 Detailed Description 11
4.5.2 Member Data Documentation 11
4.5.2.1 handle 11
4.5.2.2 msg_id 11
4.5.2.3 sd 11
4.5.2.4 sep_list 11
4.5.2.5 trans 11
4.6 coap_server_path Struct Reference 12
4.6.1 Detailed Description 12
4.6.2 Member Data Documentation 12
4.6.2.1 next 12
4.6.2.2 str 12
4.7 coap_server_path_list_t Struct Reference 12
4.7.1 Detailed Description 12
4.7.2 Member Data Documentation 12
4.7.2.1 first 12
4.7.2.2 last 13
4.8 coap_server_trans Struct Reference 13
4.8.1 Detailed Description 13
4.8.2 Member Data Documentation 13
4.8.2.1 active 13
4.8.2.2 client_addr 13
4.8.2.3 client_sin 13
4.8.2.4 client_sin_len 13
4.8.2.5 last_use 14
4.8.2.6 num_retrans 14
4.8.2.7 req 14

4.8.2.8 resp 14
4.8.2.9 server 14
4.8.2.10 timeout 14
4.8.2.11 timer_fd 14
4.9 test_client_data_t Struct Reference 14
4.9.1 Detailed Description 15
4.9.2 Member Data Documentation 15
4.9.2.1 cert_file_name 15
4.9.2.2 code_class 15
4.9.2.3 code_detail 15
4.9.2.4 crl_file_name 15
4.9.2.5 desc 15
4.9.2.6 host 15
4.9.2.7 key_file_name 15
4.9.2.8 payload 15
4.9.2.9 payload_len 15
4.9.2.10 port 15
4.9.2.11 trust_file_name 15
4.9.2.12 type 16
4.9.2.13 uri_path_opt 16
4.10 test_msg_data_t Struct Reference 16
4.10.1 Detailed Description 16
4.10.2 Member Data Documentation 16
4.10.2.1 add_op_ret 16
4.10.2.2 buf 17
4.10.2.3 buf_len 17
4.10.2.4 code_class 17
4.10.2.5 code_detail 17
4.10.2.6 copy_desc 17
4.10.2.7 copy_ret 17
4.10.2.8 format_desc 17
4.10.2.9 format_ret 17
4.10.2.10 msg_id 17
4.10.2.11 num_ops 17
4.10.2.12 ops 17
4.10.2.13 parse_desc 17
4.10.2.14 parse_ret 18
4.10.2.15 payload 18
4.10.2.16 payload_len 18
4.10.2.17 set_code_ret 18

4.10.2.18 set_msg_id_ret 18
4.10.2.19 set_payload_ret 18
4.10.2.20 set_token_ret 18
4.10.2.21 set_type_ret 18
4.10.2.22 token 18
4.10.2.23 token_len 18
4.10.2.24 type 18
4.10.2.25 ver 18
4.11 test_msg_op_t Struct Reference 19
4.11.1 Detailed Description 19
4.11.2 Member Data Documentation 19
4.11.2.1 len 19
4.11.2.2 num 19
4.11.2.3 val 19
4.12 test_t Struct Reference 19
4.12.1 Detailed Description 19
4.12.2 Member Data Documentation 20
4.12.2.1 data 20
4.12.2.2 func 20

5 File Documentation 21

5.1 include/coap_client.h File Reference 21
5.1.1 Detailed Description 21
5.1.2 Macro Definition Documentation 21
5.1.2.1 COAP_CLIENT_ADDR_BUF_LEN 21
5.1.3 Function Documentation 22
5.1.3.1 coap_client_create(coap_client_t ∗client, const char ∗host, unsigned port) 22
5.1.3.2 coap_client_destroy(coap_client_t ∗client) 22
5.1.3.3 coap_client_exchange(coap_client_t ∗client, coap_msg_t ∗req, coap_msg_t ∗resp) 22
5.2 include/coap_log.h File Reference 22
5.2.1 Detailed Description 23
5.2.2 Macro Definition Documentation 23
5.2.2.1 COAP_LOG_DEF_LEVEL 23
5.2.3 Enumeration Type Documentation 23
5.2.3.1 coap_log_level_t 23
5.2.4 Function Documentation 23
5.2.4.1 coap_log_debug(const char ∗msg,...) 23
5.2.4.2 coap_log_error(const char ∗msg,...) 24
5.2.4.3 coap_log_get_level(void) 24
5.2.4.4 coap_log_info(const char ∗msg,...) 24

5.2.4.5 coap_log_notice(const char ∗msg,...) 24
5.2.4.6 coap_log_set_level(coap_log_level_t level) 24
5.2.4.7 coap_log_warn(const char ∗msg,...) 24
5.3 include/coap_msg.h File Reference 25
5.3.1 Detailed Description 27
5.3.2 Macro Definition Documentation 27
5.3.2.1 coap_msg_get_code_class 27
5.3.2.2 coap_msg_get_code_detail 27
5.3.2.3 coap_msg_get_first_op 27
5.3.2.4 coap_msg_get_msg_id 27
5.3.2.5 coap_msg_get_payload 27
5.3.2.6 coap_msg_get_payload_len 27
5.3.2.7 coap_msg_get_token 27
5.3.2.8 coap_msg_get_token_len 27
5.3.2.9 coap_msg_get_type 28
5.3.2.10 coap_msg_get_ver 28
5.3.2.11 coap_msg_is_empty 28
5.3.2.12 COAP_MSG_MAX_BUF_LEN 28
5.3.2.13 COAP_MSG_MAX_CODE_CLASS 28
5.3.2.14 COAP_MSG_MAX_CODE_DETAIL 28
5.3.2.15 COAP_MSG_MAX_MSG_ID 28
5.3.2.16 COAP_MSG_MAX_TOKEN_LEN 28
5.3.2.17 coap_msg_op_get_len 28
5.3.2.18 coap_msg_op_get_next 28
5.3.2.19 coap_msg_op_get_num 28
5.3.2.20 coap_msg_op_get_val 28
5.3.2.21 coap_msg_op_num_is_critical 29
5.3.2.22 coap_msg_op_num_is_unsafe 29
5.3.2.23 coap_msg_op_num_no_cache_key 29
5.3.2.24 coap_msg_op_set_len 29
5.3.2.25 coap_msg_op_set_next 29
5.3.2.26 coap_msg_op_set_num 29
5.3.2.27 coap_msg_op_set_val 29
5.3.2.28 COAP_MSG_OP_URI_PATH_MAX_LEN 29
5.3.2.29 COAP_MSG_OP_URI_PATH_NUM 29
5.3.2.30 COAP_MSG_VER 29
5.3.3 Function Documentation 29
5.3.3.1 coap_msg_add_op(coap_msg_t ∗msg, unsigned num, unsigned len, const char ∗val) 29
5.3.3.2 coap_msg_copy(coap_msg_t ∗dst, coap_msg_t ∗src) 30

5.3.3.3 coap_msg_create(coap_msg_t ∗msg) 30
5.3.3.4 coap_msg_destroy(coap_msg_t ∗msg) 30
5.3.3.5 coap_msg_format(coap_msg_t ∗msg, char ∗buf, unsigned len) 30
5.3.3.6 coap_msg_gen_rand_str(char ∗buf, unsigned len) 31
5.3.3.7 coap_msg_parse(coap_msg_t ∗msg, char ∗buf, unsigned len) 31
5.3.3.8 coap_msg_parse_type_msg_id(char ∗buf, unsigned len, unsigned ∗type, unsigned ∗msg_id) 31
5.3.3.9 coap_msg_reset(coap_msg_t ∗msg) 32
5.3.3.10 coap_msg_set_code(coap_msg_t ∗msg, unsigned code_class, unsigned code←_detail) 32
5.3.3.11 coap_msg_set_msg_id(coap_msg_t ∗msg, unsigned msg_id) 32
5.3.3.12 coap_msg_set_payload(coap_msg_t ∗msg, char ∗buf, unsigned len) 33
5.3.3.13 coap_msg_set_token(coap_msg_t ∗msg, char ∗buf, unsigned len) 33
5.3.3.14 coap_msg_set_type(coap_msg_t ∗msg, unsigned type) 33
5.4 include/coap_server.h File Reference 34
5.4.1 Detailed Description 35
5.4.2 Macro Definition Documentation 35
5.4.2.1 COAP_SERVER_ADDR_BUF_LEN 35
5.4.2.2 COAP_SERVER_NUM_TRANS 35
5.4.3 Function Documentation 35
5.4.3.1 coap_server_add_sep_resp_uri_path(coap_server_t ∗server, const char ∗str) . . 35
5.4.3.2 coap_server_create(coap_server_t ∗server, int(∗handle)(coap_server_t ∗, coap_msg_t ∗, coap_msg_t ∗), const char ∗host, unsigned port) 35
5.4.3.3 coap_server_destroy(coap_server_t ∗server) 36
5.4.3.4 coap_server_get_next_msg_id(coap_server_t ∗server) 36
5.4.3.5 coap_server_run(coap_server_t ∗server) 36
5.5 src/coap_client.c File Reference 37
5.5.1 Detailed Description 37
5.5.2 Macro Definition Documentation 37
5.5.2.1 COAP_CLIENT_ACK_TIMEOUT_SEC 37
5.5.2.2 COAP_CLIENT_MAX_RETRANSMIT 37
5.5.2.3 COAP_CLIENT_RESP_TIMEOUT_SEC 38
5.5.3 Function Documentation 38
5.5.3.1 coap_client_create(coap_client_t ∗client, const char ∗host, unsigned port) 38
5.5.3.2 coap_client_destroy(coap_client_t ∗client) 38
5.5.3.3 coap_client_exchange(coap_client_t ∗client, coap_msg_t ∗req, coap_msg_t ∗resp) 38
5.6 src/coap_msg.c File Reference 38
5.6.1 Detailed Description 39
5.6.2 Macro Definition Documentation 40
5.6.2.1 coap_msg_op_list_get_first 40
5.6.2.2 coap_msg_op_list_get_last 40

5.6.2.3 coap_msg_op_list_is_empty 40
5.6.3 Function Documentation 40
5.6.3.1 coap_msg_add_op(coap_msg_t ∗msg, unsigned num, unsigned len, const char ∗val) 40
5.6.3.2 coap_msg_copy(coap_msg_t ∗dst, coap_msg_t ∗src) 40
5.6.3.3 coap_msg_create(coap_msg_t ∗msg) 41
5.6.3.4 coap_msg_destroy(coap_msg_t ∗msg) 41
5.6.3.5 coap_msg_format(coap_msg_t ∗msg, char ∗buf, unsigned len) 41
5.6.3.6 coap_msg_gen_rand_str(char ∗buf, unsigned len) 41
5.6.3.7 coap_msg_parse(coap_msg_t ∗msg, char ∗buf, unsigned len) 41
5.6.3.8 coap_msg_parse_type_msg_id(char ∗buf, unsigned len, unsigned ∗type, unsigned ∗msg_id) 42
5.6.3.9 coap_msg_reset(coap_msg_t ∗msg) 42
5.6.3.10 coap_msg_set_code(coap_msg_t ∗msg, unsigned code_class, unsigned code←_detail) 42
5.6.3.11 coap_msg_set_msg_id(coap_msg_t ∗msg, unsigned msg_id) 43
5.6.3.12 coap_msg_set_payload(coap_msg_t ∗msg, char ∗buf, unsigned len) 43
5.6.3.13 coap_msg_set_token(coap_msg_t ∗msg, char ∗buf, unsigned len) 43
5.6.3.14 coap_msg_set_type(coap_msg_t ∗msg, unsigned type) 44
5.7 src/coap_server.c File Reference 44
5.7.1 Detailed Description 45
5.7.2 Macro Definition Documentation 45
5.7.2.1 COAP_SERVER_ACK_TIMEOUT_SEC 45
5.7.2.2 COAP_SERVER_MAX_RETRANSMIT 45
5.7.3 Function Documentation 45
5.7.3.1 coap_server_add_sep_resp_uri_path(coap_server_t ∗server, const char ∗str) . . 45
5.7.3.2 coap_server_create(coap_server_t ∗server, int(∗handle)(coap_server_t ∗, coap_msg_t ∗, coap_msg_t ∗), const char ∗host, unsigned port) 45
5.7.3.3 coap_server_destroy(coap_server_t ∗server) 46
5.7.3.4 coap_server_get_next_msg_id(coap_server_t ∗server) 46
5.7.3.5 coap_server_run(coap_server_t ∗server) 46
5.8 test/test.c File Reference 47
5.8.1 Detailed Description 47
5.8.2 Macro Definition Documentation 47
5.8.2.1 FAIL_STR 47
5.8.2.2 PASS_STR 47
5.8.2.3 START_STR 47
5.8.3 Function Documentation 47
5.8.3.1 test_run(test_t ∗test, unsigned num_tests) 47
5.9 test/test.h File Reference 48
5.9.1 Detailed Description 48

5.9.2 Function Documentation 48
5.9.2.1 test_run(test_t ∗test, unsigned num_tests) 48

Index **51**

Chapter 1

README

========

FreeCoAP

An implementation of the CoAP protocol for GNU/Linux consisting of:

- a CoAP message parser/formatter library
- a CoAP client library
- a CoAP server library
- unit test code for the message parser/formatter
- a client test application
- a server test application

Test Instructions

To build and test the message parser/formatter

```
$ cd test_msg
$ make
$ ./test_msg
```

To build and run the server test application

```
$ cd test_server
$ make
$ ./server
```

To build and run the client test application

$ cd test_client

$ make

$ ./client

To build and run the server test application with DTLS enabled

$ cd test_server

$ make dtls=y

$ ./server

To build and run the client test application with DTLS enabled

$ cd test_client

$ make dtls=y

$ ./client

Chapter 2

Class Index

2.1 Class List

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

coap_client_t
Client structure 7

coap_msg_op
Option structure 8

coap_msg_op_list_t
Option linked-list structure 9

coap_msg_t
Message structure 9

coap_server
Server structure 11

coap_server_path
URI path structure 12

coap_server_path_list_t
URI path list structure 12

coap_server_trans
Transaction structure 13

test_client_data_t
Client test data structure 14

test_msg_data_t
Message test data structure 16

test_msg_op_t
Message option test data structure 19

test_t
Test structure 19

Chapter 3

File Index

3.1 File List

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

include/coap_client.h
Include file for the FreeCoAP client library 21

include/coap_log.h
Include file for the FreeCoAP logging module 22

include/coap_msg.h
Include file for the FreeCoAP message parser/formatter library 25

include/coap_server.h
Include file for the FreeCoAP server library 34

src/coap_client.c
Source file for the FreeCoAP client library 37

src/coap_msg.c
Source file for the FreeCoAP message parser/formatter library 38

src/coap_server.c
Source file for the FreeCoAP server library 44

test/test.c
Source file for the FreeCoAP test harness 47

test/test.h
Include file for the FreeCoAP test harness 48

Chapter 4

Class Documentation

4.1 coap_client_t Struct Reference

Client structure.

```
#include <coap_client.h>
```

Public Attributes

- int sd
- int timer_fd
- struct timespec timeout
- unsigned num_retrans
- struct sockaddr_in6 server_sin
- socklen_t server_sin_len
- char server_addr [COAP_CLIENT_ADDR_BUF_LEN]

4.1.1 Detailed Description

Client structure.

4.1.2 Member Data Documentation

4.1.2.1 unsigned coap_client_t::num_retrans

Current number of retransmissions

4.1.2.2 int coap_client_t::sd

Socket descriptor

4.1.2.3 char coap_client_t::server_addr[COAP_CLIENT_ADDR_BUF_LEN]

String to hold the server address

4.1.2.4 struct sockaddr_in6 coap_client_t::server_sin

Ipv6 socket structture

4.1.2.5 socklen_t coap_client_t::server_sin_len

Ipv6 socket structure length

4.1.2.6 struct timespec coap_client_t::timeout

Timeout value

4.1.2.7 int coap_client_t::timer_fd

Timer file descriptor

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

- include/coap_client.h

4.2 coap_msg_op Struct Reference

Option structure.

```
#include <coap_msg.h>
```

Public Attributes

- unsigned num
- unsigned len
- char ∗ val
- struct coap_msg_op ∗ next

4.2.1 Detailed Description

Option structure.

4.2.2 Member Data Documentation

4.2.2.1 unsigned coap_msg_op::len

Option lenght

4.2.2.2 struct coap_msg_op∗ coap_msg_op::next

Pointer to the next option structure in the list

4.2.2.3 unsigned coap_msg_op::num

Option number

4.2.2.4 char∗ coap_msg_op::val

Pointer to a buffer containing the option value

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

- include/coap_msg.h

4.3 coap_msg_op_list_t Struct Reference

Option linked-list structure.

```
#include <coap_msg.h>
```

Public Attributes

- coap_msg_op_t ∗ first
- coap_msg_op_t ∗ last

4.3.1 Detailed Description

Option linked-list structure.

4.3.2 Member Data Documentation

4.3.2.1 coap_msg_op_t∗ coap_msg_op_list_t::first

Pointer to the first option structure in the list

4.3.2.2 coap_msg_op_t∗ coap_msg_op_list_t::last

Pointer to the last option structure in the list

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

- include/coap_msg.h

4.4 coap_msg_t Struct Reference

Message structure.

```
#include <coap_msg.h>
```

Public Attributes

- unsigned ver
- coap_msg_type_t type
- unsigned token_len
- unsigned code_class
- unsigned code_detail
- unsigned msg_id
- char token [COAP_MSG_MAX_TOKEN_LEN]

- coap_msg_op_list_t op_list
- char ∗ payload
- unsigned payload_len

4.4.1 Detailed Description

Message structure.

4.4.2 Member Data Documentation

4.4.2.1 unsigned coap_msg_t::code_class

Code class

4.4.2.2 unsigned coap_msg_t::code_detail

Code detail

4.4.2.3 unsigned coap_msg_t::msg_id

Message ID

4.4.2.4 coap_msg_op_list_t coap_msg_t::op_list

Option list

4.4.2.5 char∗ coap_msg_t::payload

Pointer to a buffer containing the payload

4.4.2.6 unsigned coap_msg_t::payload_len

Length of the payload

4.4.2.7 char coap_msg_t::token[COAP_MSG_MAX_TOKEN_LEN]

Token value

4.4.2.8 unsigned coap_msg_t::token_len

Token length

4.4.2.9 coap_msg_type_t coap_msg_t::type

Message type

4.4.2.10 unsigned coap_msg_t::ver

CoAP version

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

- include/coap_msg.h

4.5 coap_server Struct Reference

Server structure.

```
#include <coap_server.h>
```

Public Attributes

- int sd
- unsigned msg_id
- coap_server_path_list_t sep_list
- coap_server_trans_t trans [COAP_SERVER_NUM_TRANS]
- int(∗ handle)(struct coap_server ∗, coap_msg_t ∗, coap_msg_t ∗)

4.5.1 Detailed Description

Server structure.

4.5.2 Member Data Documentation

4.5.2.1 int(∗ coap_server::handle) (struct coap_server ∗, coap_msg_t ∗, coap_msg_t ∗)

Call-back function to handle requests and generate responses

4.5.2.2 unsigned coap_server::msg_id

Last message ID value used in a response message

4.5.2.3 int coap_server::sd

Socket descriptor

4.5.2.4 coap_server_path_list_t coap_server::sep_list

List of URI paths that require separate responses

4.5.2.5 coap_server_trans_t coap_server::trans[COAP_SERVER_NUM_TRANS]

Array of transaction structures

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

- include/coap_server.h

4.6 coap_server_path Struct Reference

URI path structure.

```
#include <coap_server.h>
```

Public Attributes

- char ∗ str
- struct coap_server_path ∗ next

4.6.1 Detailed Description

URI path structure.

4.6.2 Member Data Documentation

4.6.2.1 struct **coap_server_path**∗ coap_server_path::next

Pointer to the next URI path structure in the list

4.6.2.2 char∗ coap_server_path::str

String containing a path

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

- include/coap_server.h

4.7 coap_server_path_list_t Struct Reference

URI path list structure.

```
#include <coap_server.h>
```

Public Attributes

- coap_server_path_t ∗ first
- coap_server_path_t ∗ last

4.7.1 Detailed Description

URI path list structure.

4.7.2 Member Data Documentation

4.7.2.1 **coap_server_path_t**∗ coap_server_path_list_t::first

Pointer to the first URI path structure in the list

4.7.2.2 coap_server_path_t∗ coap_server_path_list_t::last

Pointer to the last URI path structure in the list

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

- include/coap_server.h

4.8 coap_server_trans Struct Reference

Transaction structure.

```
#include <coap_server.h>
```

Public Attributes

- int active
- time_t last_use
- int timer_fd
- struct timespec timeout
- unsigned num_retrans
- struct sockaddr_in6 client_sin
- socklen_t client_sin_len
- char client_addr [COAP_SERVER_ADDR_BUF_LEN]
- coap_msg_t req
- coap_msg_t resp
- struct coap_server ∗ server

4.8.1 Detailed Description

Transaction structure.

4.8.2 Member Data Documentation

4.8.2.1 int coap_server_trans::active

Flag to indicate if this transaction structure contains valid data

4.8.2.2 char coap_server_trans::client_addr[COAP_SERVER_ADDR_BUF_LEN]

String to hold the client address

4.8.2.3 struct sockaddr_in6 coap_server_trans::client_sin

Ipv6 socket structure

4.8.2.4 socklen_t coap_server_trans::client_sin_len

Ipv6 socket structure length

4.8.2.5 time_t coap_server_trans::last_use

The time that this transaction structure was last used

4.8.2.6 unsigned coap_server_trans::num_retrans

Current number of retransmissions

4.8.2.7 coap_msg_t coap_server_trans::req

Last request message received for this transaction

4.8.2.8 coap_msg_t coap_server_trans::resp

Last response message sent for this transaction

4.8.2.9 struct coap_server∗ coap_server_trans::server

Pointer to the containing server structure

4.8.2.10 struct timespec coap_server_trans::timeout

Timeout value

4.8.2.11 int coap_server_trans::timer_fd

Timer file descriptor

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

- include/coap_server.h

4.9 test_client_data_t Struct Reference

Client test data structure.

Public Attributes

- const char ∗ desc
- const char ∗ host
- unsigned port
- const char ∗ key_file_name
- const char ∗ cert_file_name
- const char ∗ trust_file_name
- const char ∗ crl_file_name
- const char ∗ uri_path_opt
- coap_msg_type_t type
- unsigned code_class
- unsigned code_detail
- char ∗ payload
- unsigned payload_len

4.9.1 Detailed Description

Client test data structure.

4.9.2 Member Data Documentation

4.9.2.1 const char∗ test_client_data_t::cert_file_name

DTLS certificate file name

4.9.2.2 unsigned test_client_data_t::code_class

Message code class

4.9.2.3 unsigned test_client_data_t::code_detail

Message code detail

4.9.2.4 const char∗ test_client_data_t::crl_file_name

DTLS certificate revocation list file name

4.9.2.5 const char∗ test_client_data_t::desc

Test description

4.9.2.6 const char∗ test_client_data_t::host

Server host address

4.9.2.7 const char∗ test_client_data_t::key_file_name

DTLS key file name

4.9.2.8 char∗ test_client_data_t::payload

Buffer containing the payload

4.9.2.9 unsigned test_client_data_t::payload_len

Length of the buffer containing the payload

4.9.2.10 unsigned test_client_data_t::port

Server UDP port

4.9.2.11 const char∗ test_client_data_t::trust_file_name

DTLS trust file name

4.9.2.12 coap_msg_type_t test_client_data_t::type

Message type

4.9.2.13 const char∗ test_client_data_t::uri_path_opt

URI path option value

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

- test_client/client.c

4.10 test_msg_data_t Struct Reference

Message test data structure.

Public Attributes

- const char ∗ parse_desc
- const char ∗ format_desc
- const char ∗ copy_desc
- int parse_ret
- int set_type_ret
- int set_code_ret
- int set_msg_id_ret
- int set_token_ret
- int ∗ add_op_ret
- int set_payload_ret
- int format_ret
- int copy_ret
- char ∗ buf
- unsigned buf_len
- unsigned ver
- coap_msg_type_t type
- unsigned code_class
- unsigned code_detail
- unsigned msg_id
- char ∗ token
- unsigned token_len
- test_msg_op_t ∗ ops
- unsigned num_ops
- char ∗ payload
- unsigned payload_len

4.10.1 Detailed Description

Message test data structure.

4.10.2 Member Data Documentation

4.10.2.1 int∗ test_msg_data_t::add_op_ret

Expected return value for the add option function

4.10.2.2 char∗ test_msg_data_t::buf

Buffer containing a message

4.10.2.3 unsigned test_msg_data_t::buf_len

Length of the buffer containing a message

4.10.2.4 unsigned test_msg_data_t::code_class

Message code class

4.10.2.5 unsigned test_msg_data_t::code_detail

Message code detail

4.10.2.6 const char∗ test_msg_data_t::copy_desc

Test description for the copy test

4.10.2.7 int test_msg_data_t::copy_ret

Expected return value for the copy function

4.10.2.8 const char∗ test_msg_data_t::format_desc

Test description for the format test

4.10.2.9 int test_msg_data_t::format_ret

Expected return value for the format function

4.10.2.10 unsigned test_msg_data_t::msg_id

Message ID

4.10.2.11 unsigned test_msg_data_t::num_ops

Size of the array of message option test data structures

4.10.2.12 test_msg_op_t∗ test_msg_data_t::ops

Array of message option test data structures

4.10.2.13 const char∗ test_msg_data_t::parse_desc

Test description for the parse test

4.10.2.14 int test_msg_data_t::parse_ret

Expected return value for the parse function

4.10.2.15 char∗ test_msg_data_t::payload

Buffer containing a payload

4.10.2.16 unsigned test_msg_data_t::payload_len

Length of the buffer containing a payload

4.10.2.17 int test_msg_data_t::set_code_ret

Expected return value for the set code function

4.10.2.18 int test_msg_data_t::set_msg_id_ret

Expected return value for the set message ID function

4.10.2.19 int test_msg_data_t::set_payload_ret

Expected return value for the set payload function

4.10.2.20 int test_msg_data_t::set_token_ret

Expected return value for the set token function

4.10.2.21 int test_msg_data_t::set_type_ret

Expected return value for the set type function

4.10.2.22 char∗ test_msg_data_t::token

Buffer containing a token

4.10.2.23 unsigned test_msg_data_t::token_len

Length of the buffer containing a token

4.10.2.24 coap_msg_type_t test_msg_data_t::type

Message type

4.10.2.25 unsigned test_msg_data_t::ver

CoAP version

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

- test_msg/test_msg.c

4.11 test_msg_op_t Struct Reference

Message option test data structure.

Public Attributes

- unsigned num
- unsigned len
- char ∗ val

4.11.1 Detailed Description

Message option test data structure.

4.11.2 Member Data Documentation

4.11.2.1 unsigned test_msg_op_t::len

Option length

4.11.2.2 unsigned test_msg_op_t::num

Option number

4.11.2.3 char∗ test_msg_op_t::val

Pointer to a buffer containing the option value

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

- test_msg/test_msg.c

4.12 test_t Struct Reference

Test structure.

```
#include <test.h>
```

Public Attributes

- test_func_t func
- test_data_t data

4.12.1 Detailed Description

Test structure.

4.12.2 Member Data Documentation

4.12.2.1 test_data_t test_t::data

Test data

4.12.2.2 test_func_t test_t::func

Test function

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

- test/test.h

Chapter 5

File Documentation

5.1 include/coap_client.h File Reference

Include file for the FreeCoAP client library.

```
#include <time.h>
#include <netinet/in.h>
#include "coap_msg.h"
```

Classes

- struct coap_client_t

  Client structure.

Macros

- #define COAP_CLIENT_ADDR_BUF_LEN 128

Functions

- int coap_client_create (coap_client_t ∗client, const char ∗host, unsigned port)

  Initialise a client structure.
- void coap_client_destroy (coap_client_t ∗client)

  Deinitialise a client structure.
- int coap_client_exchange (coap_client_t ∗client, coap_msg_t ∗req, coap_msg_t ∗resp)

  Send a request to the server and receive the response.

5.1.1 Detailed Description

Include file for the FreeCoAP client library.

5.1.2 Macro Definition Documentation

5.1.2.1 #define COAP_CLIENT_ADDR_BUF_LEN 128

Buffer length for host addresses

5.1.3 Function Documentation

5.1.3.1 int coap_client_create (coap_client_t ∗ *client,* const char ∗ *host,* unsigned *port*)

Initialise a client structure.

Parameters

out	*client*	Pointer to a client structure
in	*host*	Pointer to a string containing the host address of the server
in	*port*	Port number of the server

Returns

Operation status

Return values

0	Success
<0	Error

5.1.3.2 void coap_client_destroy (coap_client_t ∗ *client*)

Deinitialise a client structure.

Parameters

in,out	*client*	Pointer to a client structure

5.1.3.3 int coap_client_exchange (coap_client_t ∗ *client,* coap_msg_t ∗ *req,* coap_msg_t ∗ *resp*)

Send a request to the server and receive the response.

Parameters

in,out	*client*	Pointer to a client structure
in	*req*	Pointer to the request message
out	*resp*	Pointer to the response message

This function sets the message ID and token fields of the request message overriding any values set by the calling function.

Returns

Operation status

Return values

0	Success
<0	Error

5.2 include/coap_log.h File Reference

Include file for the FreeCoAP logging module.

Macros

- #define COAP_LOG_DEF_LEVEL COAP_LOG_ERROR

Enumerations

- enum coap_log_level_t {
COAP_LOG_ERROR = 0, **COAP_LOG_WARN** = 1, **COAP_LOG_NOTICE** = 2, **COAP_LOG_INFO** = 3,
COAP_LOG_DEBUG = 4 }

    Log level.

Functions

- void coap_log_set_level (coap_log_level_t level)

    Set the log level.
- coap_log_level_t coap_log_get_level (void)

    Get the log level.
- void coap_log_error (const char ∗msg,...)

    Log an error message.
- void coap_log_warn (const char ∗msg,...)

    Log a warning message.
- void coap_log_notice (const char ∗msg,...)

    Log an notice message.
- void coap_log_info (const char ∗msg,...)

    Log an info message.
- void coap_log_debug (const char ∗msg,...)

    Log a debug message.

5.2.1 Detailed Description

Include file for the FreeCoAP logging module.

Source file for the FreeCoAP logging module.

5.2.2 Macro Definition Documentation

5.2.2.1 #define COAP_LOG_DEF_LEVEL **COAP_LOG_ERROR**

Default log level

5.2.3 Enumeration Type Documentation

5.2.3.1 enum **coap_log_level_t**

Log level.

Enumerator

COAP_LOG_ERROR	Highest severity level
COAP_LOG_DEBUG	Lowest severity level

5.2.4 Function Documentation

5.2.4.1 void coap_log_debug (const char ∗ *msg*, ...)

Log a debug message.

Parameters

in	*msg*	String containing format specifiers
in	...	arguments for the format specifiers

5.2.4.2 void coap_log_error (const char ∗ *msg,* *...*)

Log an error message.

Parameters

in	*msg*	String containing format specifiers
in	...	arguments for the format specifiers

5.2.4.3 coap_log_level_t coap_log_get_level (void)

Get the log level.

Returns

The current log level

5.2.4.4 void coap_log_info (const char ∗ *msg,* *...*)

Log an info message.

Parameters

in	*msg*	String containing format specifiers
in	...	arguments for the format specifiers

5.2.4.5 void coap_log_notice (const char ∗ *msg,* *...*)

Log an notice message.

Parameters

in	*msg*	String containing format specifiers
in	...	arguments for the format specifiers

5.2.4.6 void coap_log_set_level (coap_log_level_t *level*)

Set the log level.

Messages with a severity below this level will be filtered. Error messages cannot be filtered.

Parameters

in	*level*	The new log level

5.2.4.7 void coap_log_warn (const char ∗ *msg,* *...*)

Log a warning message.

Parameters

in	*msg*	String containing format specifiers
in	...	arguments for the format specifiers

5.3 include/coap_msg.h File Reference

Include file for the FreeCoAP message parser/formatter library.

Classes

- struct coap_msg_op

  Option structure.
- struct coap_msg_op_list_t

  Option linked-list structure.
- struct coap_msg_t

  Message structure.

Macros

- #define COAP_MSG_VER 0x01
- #define COAP_MSG_MAX_TOKEN_LEN 8
- #define COAP_MSG_MAX_CODE_CLASS 7
- #define COAP_MSG_MAX_CODE_DETAIL 31
- #define COAP_MSG_MAX_MSG_ID ((1 << 16) - 1)
- #define COAP_MSG_OP_URI_PATH_NUM 11
- #define COAP_MSG_OP_URI_PATH_MAX_LEN 256
- #define COAP_MSG_MAX_BUF_LEN 1152
- #define coap_msg_op_num_is_critical(num) ((num) & 1)
- #define coap_msg_op_num_is_unsafe(num) ((num) & 2)
- #define coap_msg_op_num_no_cache_key(num) ((num & 0x1e) == 0x1c)
- #define coap_msg_op_get_num(op) ((op)->num)
- #define coap_msg_op_set_num(op, num) ((op)->num = (num))
- #define coap_msg_op_get_len(op) ((op)->len)
- #define coap_msg_op_set_len(op, len) ((op)->len = (len))
- #define coap_msg_op_get_val(op) ((op)->val)
- #define coap_msg_op_set_val(op, val) ((op)->val = (val))
- #define coap_msg_op_get_next(op) ((op)->next)
- #define coap_msg_op_set_next(op, next_op) ((op)->next = (next_op))
- #define coap_msg_get_ver(msg) ((msg)->ver)
- #define coap_msg_get_type(msg) ((msg)->type)
- #define coap_msg_get_token_len(msg) ((msg)->token_len)
- #define coap_msg_get_code_class(msg) ((msg)->code_class)
- #define coap_msg_get_code_detail(msg) ((msg)->code_detail)
- #define coap_msg_get_msg_id(msg) ((msg)->msg_id)
- #define coap_msg_get_token(msg) ((msg)->token)
- #define coap_msg_get_first_op(msg) ((msg)->op_list.first)
- #define coap_msg_get_payload(msg) ((msg)->payload)
- #define coap_msg_get_payload_len(msg) ((msg)->payload_len)
- #define coap_msg_is_empty(msg) (((msg)->code_class == 0) && ((msg)->code_detail == 0))

Typedefs

- typedef struct coap_msg_op coap_msg_op_t

  Option structure.

Enumerations

- enum coap_msg_type_t { **COAP_MSG_CON** = 0x0, **COAP_MSG_NON** = 0x1, **COAP_MSG_ACK** = 0x2, **COAP_MSG_RST** = 0x3 }

  Message type enumeration.
- enum coap_msg_class_t { **COAP_MSG_REQ** = 0, **COAP_MSG_SUCCESS** = 2, **COAP_MSG_CLIENT_↩ ERR** = 4, **COAP_MSG_SERVER_ERR** = 5 }

  Code class enumeration.
- enum coap_msg_method_t { **COAP_MSG_GET** = 1, **COAP_MSG_POST** = 2, **COAP_MSG_PUT** = 3, **CO↩ AP_MSG_DELETE** =4 }

  Code detail enumeration.
- enum coap_msg_success_t {
  COAP_MSG_CREATED = 1, **COAP_MSG_DELETED** = 2, **COAP_MSG_VALID** = 3, **COAP_MSG_CHA↩ NGED** = 4,
  COAP_MSG_CONTENT = 5 }

  Success response code detail enumeration.
- enum coap_msg_client_err_t {
  COAP_MSG_BAD_REQ = 0, **COAP_MSG_UNAUTHORIZED** = 1, **COAP_MSG_BAD_OPTION** = 2, **CO↩ AP_MSG_FORBIDDEN** = 3,
  COAP_MSG_NOT_FOUND = 4, **COAP_MSG_METHOD_NOT_ALLOWED** = 5, **COAP_MSG_NOT_AC↩ CEPTABLE** = 6, **COAP_MSG_PRECOND_FAILED** = 12,
  COAP_MSG_REQ_ENT_TOO_LARGE = 13, **COAP_MSG_UNSUP_CONT_FMT** = 15 }

  Client error response code detail enumeration.
- enum coap_msg_server_err_t {
  COAP_MSG_INT_SERVER_ERR = 0, **COAP_MSG_NOT_IMPL** = 1, **COAP_MSG_BAD_GATEWAY** = 2, **COAP_MSG_SERV_UNAVAIL** = 3,
  COAP_MSG_GATEWAY_TIMEOUT = 4, **COAP_MSG_PROXY_NOT_SUP** = 5 }

  Server error response code detail enumeration.

Functions

- void coap_msg_gen_rand_str (char ∗buf, unsigned len)

  Generate a random string of bytes.
- void coap_msg_create (coap_msg_t ∗msg)

  Initialise a message structure.
- void coap_msg_destroy (coap_msg_t ∗msg)

  Deinitialise a message structure.
- void coap_msg_reset (coap_msg_t ∗msg)

  Deinitialise and initialise a message structure.
- int coap_msg_parse_type_msg_id (char ∗buf, unsigned len, unsigned ∗type, unsigned ∗msg_id)

  Extract the type and message ID values from a message.
- int coap_msg_parse (coap_msg_t ∗msg, char ∗buf, unsigned len)

  Parse a message.
- int coap_msg_set_type (coap_msg_t ∗msg, unsigned type)

  Set the type in a message.
- int coap_msg_set_code (coap_msg_t ∗msg, unsigned code_class, unsigned code_detail)

  Set the code in a message.

- int coap_msg_set_msg_id (coap_msg_t ∗msg, unsigned msg_id)

  Set the message ID in a message.
- int coap_msg_set_token (coap_msg_t ∗msg, char ∗buf, unsigned len)

  Set the token in a message.
- int coap_msg_add_op (coap_msg_t ∗msg, unsigned num, unsigned len, const char ∗val)

  Add a token to a message structure.
- int coap_msg_set_payload (coap_msg_t ∗msg, char ∗buf, unsigned len)

  Set the payload in a message.
- int coap_msg_format (coap_msg_t ∗msg, char ∗buf, unsigned len)

  Format a message.
- int coap_msg_copy (coap_msg_t ∗dst, coap_msg_t ∗src)

  Copy a message.

5.3.1 Detailed Description

Include file for the FreeCoAP message parser/formatter library.

5.3.2 Macro Definition Documentation

5.3.2.1 #define coap_msg_get_code_class(*msg*) ((msg)->code_class)

Get the code class from a message

5.3.2.2 #define coap_msg_get_code_detail(*msg*) ((msg)->code_detail)

Get the code detail from a message

5.3.2.3 #define coap_msg_get_first_op(*msg*) ((msg)->op_list.first)

Get the first option from a message

5.3.2.4 #define coap_msg_get_msg_id(*msg*) ((msg)->msg_id)

Get the message ID from message

5.3.2.5 #define coap_msg_get_payload(*msg*) ((msg)->payload)

Get the payload from a message

5.3.2.6 #define coap_msg_get_payload_len(*msg*) ((msg)->payload_len)

Get the payload length from a message

5.3.2.7 #define coap_msg_get_token(*msg*) ((msg)->token)

Get the token from a message

5.3.2.8 #define coap_msg_get_token_len(*msg*) ((msg)->token_len)

Get the token length from a message

5.3.2.9 #define coap_msg_get_type(*msg*) ((msg)->type)

Get the type from a message

5.3.2.10 #define coap_msg_get_ver(*msg*) ((msg)->ver)

Get the version from a message

5.3.2.11 #define coap_msg_is_empty(*msg*) (((msg)->code_class == 0) && ((msg)->code_detail == 0))

Indicate if a message is empty

5.3.2.12 #define COAP_MSG_MAX_BUF_LEN 1152

Maximum buffer length for header and payload

5.3.2.13 #define COAP_MSG_MAX_CODE_CLASS 7

Maximum code class

5.3.2.14 #define COAP_MSG_MAX_CODE_DETAIL 31

Maximum code detail

5.3.2.15 #define COAP_MSG_MAX_MSG_ID ((1 << 16) - 1)

Maximum message ID

5.3.2.16 #define COAP_MSG_MAX_TOKEN_LEN 8

Maximum token length

5.3.2.17 #define coap_msg_op_get_len(*op*) ((op)->len)

Get the option length from an option

5.3.2.18 #define coap_msg_op_get_next(*op*) ((op)->next)

Get the next pointer from an option

5.3.2.19 #define coap_msg_op_get_num(*op*) ((op)->num)

Get the option number from an option

5.3.2.20 #define coap_msg_op_get_val(*op*) ((op)->val)

Get the option value from an option

5.3.2.21 #define coap_msg_op_num_is_critical(*num*) ((num) & 1)

Indicate if an option is critical

5.3.2.22 #define coap_msg_op_num_is_unsafe(*num*) ((num) & 2)

Indicate if an option is unsafe to forward

5.3.2.23 #define coap_msg_op_num_no_cache_key(*num*) ((num & 0x1e) == 0x1c)

Indicate if an option is not part of the cache key

5.3.2.24 #define coap_msg_op_set_len(*op,* *len*) ((op)->len = (len))

Set the option length in an option

5.3.2.25 #define coap_msg_op_set_next(*op,* *next_op*) ((op)->next = (next_op))

Set the next pointer in an option

5.3.2.26 #define coap_msg_op_set_num(*op,* *num*) ((op)->num = (num))

Set the option number in an option

5.3.2.27 #define coap_msg_op_set_val(*op,* *val*) ((op)->val = (val))

Set the option value in an option

5.3.2.28 #define COAP_MSG_OP_URI_PATH_MAX_LEN 256

Maximum buffer length for a reconstructed URI path

5.3.2.29 #define COAP_MSG_OP_URI_PATH_NUM 11

Uri-path option number

5.3.2.30 #define COAP_MSG_VER 0x01

CoAP version

5.3.3 Function Documentation

5.3.3.1 int coap_msg_add_op (coap_msg_t ∗ *msg,* unsigned *num,* unsigned *len,* const char ∗ *val*)

Add a token to a message structure.

Parameters

in,out	*msg*	Pointer to a message structure
in	*num*	Option number
in	*len*	Option length
in	*val*	Pointer to a buffer containing the option value

Returns

Operation status

Return values

0	Success
<0	Error

5.3.3.2 int coap_msg_copy (coap_msg_t * *dst,* coap_msg_t * *src*)

Copy a message.

Parameters

in,out	*dst*	Pointer to the destination message structure
in	*src*	Pointer to the source message structure

Returns

Operation status

Return values

0	Success
<0	Error

5.3.3.3 void coap_msg_create (coap_msg_t * *msg*)

Initialise a message structure.

Parameters

out	*msg*	Pointer to a message structure

5.3.3.4 void coap_msg_destroy (coap_msg_t * *msg*)

Deinitialise a message structure.

Parameters

in,out	*msg*	Pointer to a message structure

5.3.3.5 int coap_msg_format (coap_msg_t * *msg,* char * *buf,* unsigned *len*)

Format a message.

Parameters

in	*msg*	Pointer to a message structure
out	*buf*	Pointer to a buffer to contain the formatted message
in	*len*	Length of the buffer

Returns

Operation status

Return values

>0	Length of the formatted message
<0	Error

5.3.3.6 void coap_msg_gen_rand_str (char * *buf*, unsigned *len*)

Generate a random string of bytes.

Parameters

out	*buf*	Pointer to the buffer to store the random string
in	*len*	Length of the buffer

5.3.3.7 int coap_msg_parse (coap_msg_t * *msg*, char * *buf*, unsigned *len*)

Parse a message.

Parameters

in,out	*msg*	Pointer to a message structure
in	*buf*	Pointer to a buffer containing the message
in	*len*	Length of the buffer

Returns

Operation status

Return values

0	Success
<0	Error

5.3.3.8 int coap_msg_parse_type_msg_id (char * *buf*, unsigned *len*, unsigned * *type*, unsigned * *msg_id*)

Extract the type and message ID values from a message.

If a message contains a format error, this function will attempt to extract the type and message ID so that a reset message can be returned to the sender.

Parameters

in	*buf*	Pointer to a buffer containing the message

in	*len*	Length of the buffer
out	*type*	Pointer to field to store the type value
out	*msg_id*	Pointer to a field to store the message ID value

Returns

Operation status

Return values

0	Success
<*0*	Error

5.3.3.9 void coap_msg_reset (coap_msg_t ∗ *msg*)

Deinitialise and initialise a message structure.

Parameters

in,out	*msg*	Pointer to a message structure

5.3.3.10 int coap_msg_set_code (coap_msg_t ∗ *msg,* unsigned *code_class,* unsigned *code_detail*)

Set the code in a message.

Parameters

out	*msg*	Pointer to a message structure
in	*code_class*	Code class
in	*code_detail*	Code detail

Returns

Operation status

Return values

0	Success
<*0*	Error

5.3.3.11 int coap_msg_set_msg_id (coap_msg_t ∗ *msg,* unsigned *msg_id*)

Set the message ID in a message.

Parameters

out	*msg*	Pointer to a message structure
in	*msg_id*	Message ID

Returns

Operation status

Return values

0	Success
<0	Error

5.3.3.12 int coap_msg_set_payload (coap_msg_t ∗ *msg,* char ∗ *buf,* unsigned *len*)

Set the payload in a message.

Free the buffer in the message structure containing the current payload if there is one, allocate a buffer to contain the new payload and copy the buffer argument into the new payload buffer.

Parameters

`in,out`	*msg*	Pointer to a message structure
`in`	*buf*	Pointer to a buffer containing the payload
`in`	*len*	Length of the buffer

Returns

Operation status

Return values

0	Success
<0	Error

5.3.3.13 int coap_msg_set_token (coap_msg_t ∗ *msg,* char ∗ *buf,* unsigned *len*)

Set the token in a message.

Parameters

`out`	*msg*	Pointer to a message structure
`in`	*buf*	Pointer to a buffer containing the token
`in`	*len*	Length of the buffer

Returns

Operation status

Return values

0	Success
<0	Error

5.3.3.14 int coap_msg_set_type (coap_msg_t ∗ *msg,* unsigned *type*)

Set the type in a message.

Parameters

`out`	*msg*	Pointer to a message structure

in	*type*	Message type

Returns

Operation status

Return values

0	Success
<*0*	Error

5.4 include/coap_server.h File Reference

Include file for the FreeCoAP server library.

```
#include <time.h>
#include <netinet/in.h>
#include "coap_msg.h"
```

Classes

- struct coap_server_path

    URI path structure.
- struct coap_server_path_list_t

    URI path list structure.
- struct coap_server_trans

    Transaction structure.
- struct coap_server

    Server structure.

Macros

- #define COAP_SERVER_NUM_TRANS 8
- #define COAP_SERVER_ADDR_BUF_LEN 128

Typedefs

- typedef struct coap_server_path coap_server_path_t

    URI path structure.
- typedef struct coap_server_trans coap_server_trans_t

    Transaction structure.
- typedef struct coap_server coap_server_t

    Server structure.

Enumerations

- enum coap_server_resp_t { **COAP_SERVER_PIGGYBACKED** = 0, **COAP_SERVER_SEPARATE** = 1 }

    Response type enumeration.

Functions

- int coap_server_create (coap_server_t ∗server, int(∗handle)(coap_server_t ∗, coap_msg_t ∗, coap_msg_t ∗), const char ∗host, unsigned port)

  Initialise a server structure.
- void coap_server_destroy (coap_server_t ∗server)

  Deinitialise a server structure.
- unsigned coap_server_get_next_msg_id (coap_server_t ∗server)

  Get a new message ID value.
- int coap_server_add_sep_resp_uri_path (coap_server_t ∗server, const char ∗str)

  Register a URI path that requires a separate response.
- int coap_server_run (coap_server_t ∗server)

  Run the server.

5.4.1 Detailed Description

Include file for the FreeCoAP server library.

5.4.2 Macro Definition Documentation

5.4.2.1 #define COAP_SERVER_ADDR_BUF_LEN 128

Buffer length for host addresses

5.4.2.2 #define COAP_SERVER_NUM_TRANS 8

Maximum number of active transactions per server

5.4.3 Function Documentation

5.4.3.1 int coap_server_add_sep_resp_uri_path (coap_server_t ∗ *server,* const char ∗ *str*)

Register a URI path that requires a separate response.

Parameters

`in,out`	*server*	Pointer to a server structure
`in`	*str*	String representation of a URI path

Returns

Operation status

Return values

0	Success
<0	Error

5.4.3.2 int coap_server_create (coap_server_t ∗ *server,* int(∗)(coap_server_t ∗, coap_msg_t ∗, coap_msg_t ∗) *handle,* const char ∗ *host,* unsigned *port*)

Initialise a server structure.

Parameters

`out`	*server*	Pointer to a server structure
`in`	*handle*	Call-back function to handle client requests
`in`	*host*	Pointer to a string containing the host address of the server
`in`	*port*	Port number of the server

Returns

Operation status

Return values

0	Success
<*0*	Error

5.4.3.3 void coap_server_destroy (coap_server_t * *server*)

Deinitialise a server structure.

Parameters

`in,out`	*server*	Pointer to a server structure

5.4.3.4 unsigned coap_server_get_next_msg_id (coap_server_t * *server*)

Get a new message ID value.

Parameters

`in,out`	*server*	Pointer to a server structure

Returns

message ID value

5.4.3.5 int coap_server_run (coap_server_t * *server*)

Run the server.

Listen for incoming requests. For each request received, call the handle call-back function in the server structure and send the response to the client.

Parameters

`in,out`	*server*	Pointer to a server structure

Returns

Operation status

Return values

0	Success
<0	Error

5.5 src/coap_client.c File Reference

Source file for the FreeCoAP client library.

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <stdint.h>
#include <errno.h>
#include <unistd.h>
#include <fcntl.h>
#include <arpa/inet.h>
#include <sys/select.h>
#include <sys/timerfd.h>
#include "coap_client.h"
#include "coap_log.h"
```

Macros

- #define COAP_CLIENT_ACK_TIMEOUT_SEC 2
- #define COAP_CLIENT_MAX_RETRANSMIT 4
- #define COAP_CLIENT_RESP_TIMEOUT_SEC 30

Functions

- int coap_client_create (coap_client_t ∗client, const char ∗host, unsigned port)

  Initialise a client structure.
- void coap_client_destroy (coap_client_t ∗client)

  Deinitialise a client structure.
- int coap_client_exchange (coap_client_t ∗client, coap_msg_t ∗req, coap_msg_t ∗resp)

  Send a request to the server and receive the response.

5.5.1 Detailed Description

Source file for the FreeCoAP client library.

5.5.2 Macro Definition Documentation

5.5.2.1 #define COAP_CLIENT_ACK_TIMEOUT_SEC 2

Minimum delay to wait before retransmitting a confirmable message

5.5.2.2 #define COAP_CLIENT_MAX_RETRANSMIT 4

Maximum number of times a confirmable message can be retransmitted

5.5.2.3 #define COAP_CLIENT_RESP_TIMEOUT_SEC 30

Maximum amount of time to wait for a response

5.5.3 Function Documentation

5.5.3.1 int coap_client_create (coap_client_t ∗ *client,* const char ∗ *host,* unsigned *port*)

Initialise a client structure.

Parameters

out	*client*	Pointer to a client structure
in	*host*	Pointer to a string containing the host address of the server
in	*port*	Port number of the server

Returns

Operation status

Return values

0	Success
<*0*	Error

5.5.3.2 void coap_client_destroy (coap_client_t ∗ *client*)

Deinitialise a client structure.

Parameters

in,out	*client*	Pointer to a client structure

5.5.3.3 int coap_client_exchange (coap_client_t ∗ *client,* coap_msg_t ∗ *req,* coap_msg_t ∗ *resp*)

Send a request to the server and receive the response.

Parameters

in,out	*client*	Pointer to a client structure
in	*req*	Pointer to the request message
out	*resp*	Pointer to the response message

This function sets the message ID and token fields of the request message overriding any values set by the calling function.

Returns

Operation status

Return values

0	Success
<*0*	Error

5.6 src/coap_msg.c File Reference

Source file for the FreeCoAP message parser/formatter library.

```
#include <stdlib.h>
#include <string.h>
#include <stdint.h>
#include <time.h>
#include <errno.h>
#include <arpa/inet.h>
#include "coap_msg.h"
```

Macros

- #define coap_msg_op_list_get_first(list) ((list)->first)
- #define coap_msg_op_list_get_last(list) ((list)->last)
- #define coap_msg_op_list_is_empty(list) ((list)->first == NULL)

Functions

- void coap_msg_gen_rand_str (char ∗buf, unsigned len)

  Generate a random string of bytes.
- void coap_msg_create (coap_msg_t ∗msg)

  Initialise a message structure.
- void coap_msg_destroy (coap_msg_t ∗msg)

  Deinitialise a message structure.
- void coap_msg_reset (coap_msg_t ∗msg)

  Deinitialise and initialise a message structure.
- int coap_msg_parse_type_msg_id (char ∗buf, unsigned len, unsigned ∗type, unsigned ∗msg_id)

  Extract the type and message ID values from a message.
- int coap_msg_parse (coap_msg_t ∗msg, char ∗buf, unsigned len)

  Parse a message.
- int coap_msg_set_type (coap_msg_t ∗msg, unsigned type)

  Set the type in a message.
- int coap_msg_set_code (coap_msg_t ∗msg, unsigned code_class, unsigned code_detail)

  Set the code in a message.
- int coap_msg_set_msg_id (coap_msg_t ∗msg, unsigned msg_id)

  Set the message ID in a message.
- int coap_msg_set_token (coap_msg_t ∗msg, char ∗buf, unsigned len)

  Set the token in a message.
- int coap_msg_add_op (coap_msg_t ∗msg, unsigned num, unsigned len, const char ∗val)

  Add a token to a message structure.
- int coap_msg_set_payload (coap_msg_t ∗msg, char ∗buf, unsigned len)

  Set the payload in a message.
- int coap_msg_format (coap_msg_t ∗msg, char ∗buf, unsigned len)

  Format a message.
- int coap_msg_copy (coap_msg_t ∗dst, coap_msg_t ∗src)

  Copy a message.

5.6.1 Detailed Description

Source file for the FreeCoAP message parser/formatter library.

5.6.2 Macro Definition Documentation

5.6.2.1 #define coap_msg_op_list_get_first(*list*) ((list)->first)

Get the first option from an option linked-list

5.6.2.2 #define coap_msg_op_list_get_last(*list*) ((list)->last)

Get the last option in an option linked-list

5.6.2.3 #define coap_msg_op_list_is_empty(*list*) ((list)->first == NULL)

Indicate if an option linked-list is empty

5.6.3 Function Documentation

5.6.3.1 int coap_msg_add_op (coap_msg_t * *msg*, unsigned *num*, unsigned *len*, const char * *val*)

Add a token to a message structure.

Parameters

`in,out`	*msg*	Pointer to a message structure
`in`	*num*	Option number
`in`	*len*	Option length
`in`	*val*	Pointer to a buffer containing the option value

Returns

Operation status

Return values

0	Success
<0	Error

5.6.3.2 int coap_msg_copy (coap_msg_t * *dst*, coap_msg_t * *src*)

Copy a message.

Parameters

`in,out`	*dst*	Pointer to the destination message structure
`in`	*src*	Pointer to the source message structure

Returns

Operation status

Return values

0	Success

<0	Error

5.6.3.3 void coap_msg_create (coap_msg_t * *msg*)

Initialise a message structure.

Parameters

out	*msg*	Pointer to a message structure

5.6.3.4 void coap_msg_destroy (coap_msg_t * *msg*)

Deinitialise a message structure.

Parameters

in,out	*msg*	Pointer to a message structure

5.6.3.5 int coap_msg_format (coap_msg_t * *msg,* char * *buf,* unsigned *len*)

Format a message.

Parameters

in	*msg*	Pointer to a message structure
out	*buf*	Pointer to a buffer to contain the formatted message
in	*len*	Length of the buffer

Returns

Operation status

Return values

>0	Length of the formatted message
<0	Error

5.6.3.6 void coap_msg_gen_rand_str (char * *buf,* unsigned *len*)

Generate a random string of bytes.

Parameters

out	*buf*	Pointer to the buffer to store the random string
in	*len*	Length of the buffer

5.6.3.7 int coap_msg_parse (coap_msg_t * *msg,* char * *buf,* unsigned *len*)

Parse a message.

Parameters

in,out	*msg*	Pointer to a message structure
in	*buf*	Pointer to a buffer containing the message
in	*len*	Length of the buffer

Returns

Operation status

Return values

0	Success
<0	Error

5.6.3.8 int coap_msg_parse_type_msg_id (char * *buf,* unsigned *len,* unsigned * *type,* unsigned * *msg_id*)

Extract the type and message ID values from a message.

If a message contains a format error, this function will attempt to extract the type and message ID so that a reset message can be returned to the sender.

Parameters

in	*buf*	Pointer to a buffer containing the message
in	*len*	Length of the buffer
out	*type*	Pointer to field to store the type value
out	*msg_id*	Pointer to a field to store the message ID value

Returns

Operation status

Return values

0	Success
<0	Error

5.6.3.9 void coap_msg_reset (coap_msg_t * *msg*)

Deinitialise and initialise a message structure.

Parameters

in,out	*msg*	Pointer to a message structure

5.6.3.10 int coap_msg_set_code (coap_msg_t * *msg,* unsigned *code_class,* unsigned *code_detail*)

Set the code in a message.

Parameters

out	*msg*	Pointer to a message structure

in	*code_class*	Code class
in	*code_detail*	Code detail

Returns

Operation status

Return values

0	Success
<*0*	Error

5.6.3.11 int coap_msg_set_msg_id (coap_msg_t ∗ *msg,* unsigned *msg_id*)

Set the message ID in a message.

Parameters

out	*msg*	Pointer to a message structure
in	*msg_id*	Message ID

Returns

Operation status

Return values

0	Success
<*0*	Error

5.6.3.12 int coap_msg_set_payload (coap_msg_t ∗ *msg,* char ∗ *buf,* unsigned *len*)

Set the payload in a message.

Free the buffer in the message structure containing the current payload if there is one, allocate a buffer to contain the new payload and copy the buffer argument into the new payload buffer.

Parameters

in,out	*msg*	Pointer to a message structure
in	*buf*	Pointer to a buffer containing the payload
in	*len*	Length of the buffer

Returns

Operation status

Return values

0	Success
<*0*	Error

5.6.3.13 int coap_msg_set_token (coap_msg_t ∗ *msg,* char ∗ *buf,* unsigned *len*)

Set the token in a message.

Parameters

out	*msg*	Pointer to a message structure
in	*buf*	Pointer to a buffer containing the token
in	*len*	Length of the buffer

Returns

Operation status

Return values

0	Success
<0	Error

5.6.3.14 int coap_msg_set_type (coap_msg_t * *msg*, unsigned *type*)

Set the type in a message.

Parameters

out	*msg*	Pointer to a message structure
in	*type*	Message type

Returns

Operation status

Return values

0	Success
<0	Error

5.7 src/coap_server.c File Reference

Source file for the FreeCoAP server library.

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <stdint.h>
#include <time.h>
#include <errno.h>
#include <unistd.h>
#include <fcntl.h>
#include <arpa/inet.h>
#include <sys/select.h>
#include <sys/timerfd.h>
#include "coap_server.h"
#include "coap_log.h"
```

Macros

- #define COAP_SERVER_ACK_TIMEOUT_SEC 2
- #define COAP_SERVER_MAX_RETRANSMIT 4

Functions

- int coap_server_create (coap_server_t ∗server, int(∗handle)(coap_server_t ∗, coap_msg_t ∗, coap_msg_t ∗), const char ∗host, unsigned port)

  Initialise a server structure.
- void coap_server_destroy (coap_server_t ∗server)

  Deinitialise a server structure.
- unsigned coap_server_get_next_msg_id (coap_server_t ∗server)

  Get a new message ID value.
- int coap_server_add_sep_resp_uri_path (coap_server_t ∗server, const char ∗str)

  Register a URI path that requires a separate response.
- int coap_server_run (coap_server_t ∗server)

  Run the server.

5.7.1 Detailed Description

Source file for the FreeCoAP server library.

5.7.2 Macro Definition Documentation

5.7.2.1 #define COAP_SERVER_ACK_TIMEOUT_SEC 2

Minimum delay to wait before retransmitting a confirmable message

5.7.2.2 #define COAP_SERVER_MAX_RETRANSMIT 4

Maximum number of times a confirmable message can be retransmitted

5.7.3 Function Documentation

5.7.3.1 int coap_server_add_sep_resp_uri_path (coap_server_t ∗ *server,* const char ∗ *str*)

Register a URI path that requires a separate response.

Parameters

`in,out`	*server*	Pointer to a server structure
`in`	*str*	String representation of a URI path

Returns

Operation status

Return values

0	Success
<0	Error

5.7.3.2 int coap_server_create (coap_server_t ∗ *server,* int(∗)(coap_server_t ∗, coap_msg_t ∗, coap_msg_t ∗) *handle,* const char ∗ *host,* unsigned *port*)

Initialise a server structure.

Parameters

out	*server*	Pointer to a server structure
in	*handle*	Call-back function to handle client requests
in	*host*	Pointer to a string containing the host address of the server
in	*port*	Port number of the server

Returns

Operation status

Return values

0	Success
<*0*	Error

5.7.3.3 void coap_server_destroy (coap_server_t * *server*)

Deinitialise a server structure.

Parameters

in,out	*server*	Pointer to a server structure

5.7.3.4 unsigned coap_server_get_next_msg_id (coap_server_t * *server*)

Get a new message ID value.

Parameters

in,out	*server*	Pointer to a server structure

Returns

message ID value

5.7.3.5 int coap_server_run (coap_server_t * *server*)

Run the server.

Listen for incoming requests. For each request received, call the handle call-back function in the server structure and send the response to the client.

Parameters

in,out	*server*	Pointer to a server structure

Returns

Operation status

Return values

0	Success
<0	Error

5.8 test/test.c File Reference

Source file for the FreeCoAP test harness.

```
#include <stdlib.h>
#include <stdio.h>
#include "test.h"
```

Macros

- #define START_STR "\n--\n"
- #define PASS_STR "-----------------<pass>-----------------\n"
- #define FAIL_STR "*****************[FAIL]*****************\n"

Functions

- unsigned test_run (test_t *test, unsigned num_tests)

    Run the tests.

5.8.1 Detailed Description

Source file for the FreeCoAP test harness.

5.8.2 Macro Definition Documentation

5.8.2.1 #define FAIL_STR "*****************[FAIL]*****************\n"

String printed at the end of a test that failed

5.8.2.2 #define PASS_STR "-----------------<pass>-----------------\n"

String printed at the end of a test that passed

5.8.2.3 #define START_STR "\n--\n"

String printed at the start of a test

5.8.3 Function Documentation

5.8.3.1 unsigned test_run (test_t * *test,* unsigned *num_tests*)

Run the tests.

Parameters

in	*test*	Pointer to an array of test structures
in	*num_tests*	Number of test structures in the array

Returns

Number of tests that passed

5.9 test/test.h File Reference

Include file for the FreeCoAP test harness.

Classes

- struct test_t

  Test structure.

Macros

- #define **DEBUG_PRINT**(fmt, ...)

Typedefs

- typedef void ∗ test_data_t

  Test data typedef.
- typedef test_result_t(∗ test_func_t) (test_data_t)

  Test function typedef.

Enumerations

- enum test_result_t { **FAIL** = 0, **PASS** }

  Test result enumeration.

Functions

- unsigned test_run (test_t ∗test, unsigned num_tests)

  Run the tests.

5.9.1 Detailed Description

Include file for the FreeCoAP test harness.

5.9.2 Function Documentation

5.9.2.1 unsigned test_run (test_t ∗ *test,* unsigned *num_tests*)

Run the tests.

Parameters

`in`	*test*	Pointer to an array of test structures
`in`	*num_tests*	Number of test structures in the array

Returns

Number of tests that passed

Index

active
    coap_server_trans, 13
add_op_ret
    test_msg_data_t, 16

buf
    test_msg_data_t, 16
buf_len
    test_msg_data_t, 17

COAP_CLIENT_ACK_TIMEOUT_SEC
    coap_client.c, 37
COAP_CLIENT_ADDR_BUF_LEN
    coap_client.h, 21
COAP_CLIENT_MAX_RETRANSMIT
    coap_client.c, 37
COAP_CLIENT_RESP_TIMEOUT_SEC
    coap_client.c, 37
COAP_LOG_DEBUG
    coap_log.h, 23
COAP_LOG_DEF_LEVEL
    coap_log.h, 23
COAP_LOG_ERROR
    coap_log.h, 23
COAP_MSG_MAX_BUF_LEN
    coap_msg.h, 28
COAP_MSG_MAX_CODE_CLASS
    coap_msg.h, 28
COAP_MSG_MAX_CODE_DETAIL
    coap_msg.h, 28
COAP_MSG_MAX_MSG_ID
    coap_msg.h, 28
COAP_MSG_MAX_TOKEN_LEN
    coap_msg.h, 28
COAP_MSG_OP_URI_PATH_MAX_LEN
    coap_msg.h, 29
COAP_MSG_OP_URI_PATH_NUM
    coap_msg.h, 29
COAP_MSG_VER
    coap_msg.h, 29
COAP_SERVER_ACK_TIMEOUT_SEC
    coap_server.c, 45
COAP_SERVER_ADDR_BUF_LEN
    coap_server.h, 35
COAP_SERVER_MAX_RETRANSMIT
    coap_server.c, 45
COAP_SERVER_NUM_TRANS
    coap_server.h, 35
cert_file_name
    test_client_data_t, 15
client_addr
    coap_server_trans, 13
client_sin
    coap_server_trans, 13
client_sin_len
    coap_server_trans, 13
coap_client.c
    COAP_CLIENT_ACK_TIMEOUT_SEC, 37
    COAP_CLIENT_MAX_RETRANSMIT, 37
    COAP_CLIENT_RESP_TIMEOUT_SEC, 37
    coap_client_create, 38
    coap_client_destroy, 38
    coap_client_exchange, 38
coap_client.h
    COAP_CLIENT_ADDR_BUF_LEN, 21
    coap_client_create, 22
    coap_client_destroy, 22
    coap_client_exchange, 22
coap_client_create
    coap_client.c, 38
    coap_client.h, 22
coap_client_destroy
    coap_client.c, 38
    coap_client.h, 22
coap_client_exchange
    coap_client.c, 38
    coap_client.h, 22
coap_client_t, 7
    num_retrans, 7
    sd, 7
    server_addr, 7
    server_sin, 7
    server_sin_len, 7
    timeout, 8
    timer_fd, 8
coap_log.h
    COAP_LOG_DEBUG, 23
    COAP_LOG_DEF_LEVEL, 23
    COAP_LOG_ERROR, 23
    coap_log_debug, 23
    coap_log_error, 24
    coap_log_get_level, 24
    coap_log_info, 24
    coap_log_level_t, 23
    coap_log_notice, 24
    coap_log_set_level, 24
    coap_log_warn, 24
coap_log_debug
    coap_log.h, 23

coap_log_error
  coap_log.h, 24
coap_log_get_level
  coap_log.h, 24
coap_log_info
  coap_log.h, 24
coap_log_level_t
  coap_log.h, 23
coap_log_notice
  coap_log.h, 24
coap_log_set_level
  coap_log.h, 24
coap_log_warn
  coap_log.h, 24
coap_msg.c
  coap_msg_add_op, 40
  coap_msg_copy, 40
  coap_msg_create, 41
  coap_msg_destroy, 41
  coap_msg_format, 41
  coap_msg_gen_rand_str, 41
  coap_msg_op_list_get_first, 40
  coap_msg_op_list_get_last, 40
  coap_msg_op_list_is_empty, 40
  coap_msg_parse, 41
  coap_msg_parse_type_msg_id, 42
  coap_msg_reset, 42
  coap_msg_set_code, 42
  coap_msg_set_msg_id, 43
  coap_msg_set_payload, 43
  coap_msg_set_token, 43
  coap_msg_set_type, 44
coap_msg.h
  COAP_MSG_MAX_BUF_LEN, 28
  COAP_MSG_MAX_CODE_CLASS, 28
  COAP_MSG_MAX_CODE_DETAIL, 28
  COAP_MSG_MAX_MSG_ID, 28
  COAP_MSG_MAX_TOKEN_LEN, 28
  COAP_MSG_OP_URI_PATH_MAX_LEN, 29
  COAP_MSG_OP_URI_PATH_NUM, 29
  COAP_MSG_VER, 29
  coap_msg_add_op, 29
  coap_msg_copy, 30
  coap_msg_create, 30
  coap_msg_destroy, 30
  coap_msg_format, 30
  coap_msg_gen_rand_str, 31
  coap_msg_get_code_class, 27
  coap_msg_get_code_detail, 27
  coap_msg_get_first_op, 27
  coap_msg_get_msg_id, 27
  coap_msg_get_payload, 27
  coap_msg_get_payload_len, 27
  coap_msg_get_token, 27
  coap_msg_get_token_len, 27
  coap_msg_get_type, 27
  coap_msg_get_ver, 28
  coap_msg_is_empty, 28
  coap_msg_op_get_len, 28
  coap_msg_op_get_next, 28
  coap_msg_op_get_num, 28
  coap_msg_op_get_val, 28
  coap_msg_op_num_is_critical, 28
  coap_msg_op_num_is_unsafe, 29
  coap_msg_op_num_no_cache_key, 29
  coap_msg_op_set_len, 29
  coap_msg_op_set_next, 29
  coap_msg_op_set_num, 29
  coap_msg_op_set_val, 29
  coap_msg_parse, 31
  coap_msg_parse_type_msg_id, 31
  coap_msg_reset, 32
  coap_msg_set_code, 32
  coap_msg_set_msg_id, 32
  coap_msg_set_payload, 33
  coap_msg_set_token, 33
  coap_msg_set_type, 33
coap_msg_add_op
  coap_msg.c, 40
  coap_msg.h, 29
coap_msg_copy
  coap_msg.c, 40
  coap_msg.h, 30
coap_msg_create
  coap_msg.c, 41
  coap_msg.h, 30
coap_msg_destroy
  coap_msg.c, 41
  coap_msg.h, 30
coap_msg_format
  coap_msg.c, 41
  coap_msg.h, 30
coap_msg_gen_rand_str
  coap_msg.c, 41
  coap_msg.h, 31
coap_msg_get_code_class
  coap_msg.h, 27
coap_msg_get_code_detail
  coap_msg.h, 27
coap_msg_get_first_op
  coap_msg.h, 27
coap_msg_get_msg_id
  coap_msg.h, 27
coap_msg_get_payload
  coap_msg.h, 27
coap_msg_get_payload_len
  coap_msg.h, 27
coap_msg_get_token
  coap_msg.h, 27
coap_msg_get_token_len
  coap_msg.h, 27
coap_msg_get_type
  coap_msg.h, 27
coap_msg_get_ver
  coap_msg.h, 28
coap_msg_is_empty

coap_msg.h, 28
coap_msg_op, 8
len, 8
next, 8
num, 8
val, 8
coap_msg_op_get_len
coap_msg.h, 28
coap_msg_op_get_next
coap_msg.h, 28
coap_msg_op_get_num
coap_msg.h, 28
coap_msg_op_get_val
coap_msg.h, 28
coap_msg_op_list_get_first
coap_msg.c, 40
coap_msg_op_list_get_last
coap_msg.c, 40
coap_msg_op_list_is_empty
coap_msg.c, 40
coap_msg_op_list_t, 9
first, 9
last, 9
coap_msg_op_num_is_critical
coap_msg.h, 28
coap_msg_op_num_is_unsafe
coap_msg.h, 29
coap_msg_op_num_no_cache_key
coap_msg.h, 29
coap_msg_op_set_len
coap_msg.h, 29
coap_msg_op_set_next
coap_msg.h, 29
coap_msg_op_set_num
coap_msg.h, 29
coap_msg_op_set_val
coap_msg.h, 29
coap_msg_parse
coap_msg.c, 41
coap_msg.h, 31
coap_msg_parse_type_msg_id
coap_msg.c, 42
coap_msg.h, 31
coap_msg_reset
coap_msg.c, 42
coap_msg.h, 32
coap_msg_set_code
coap_msg.c, 42
coap_msg.h, 32
coap_msg_set_msg_id
coap_msg.c, 43
coap_msg.h, 32
coap_msg_set_payload
coap_msg.c, 43
coap_msg.h, 33
coap_msg_set_token
coap_msg.c, 43
coap_msg.h, 33
coap_msg_set_type
coap_msg.c, 44
coap_msg.h, 33
coap_msg_t, 9
code_class, 10
code_detail, 10
msg_id, 10
op_list, 10
payload, 10
payload_len, 10
token, 10
token_len, 10
type, 10
ver, 10
coap_server, 11
handle, 11
msg_id, 11
sd, 11
sep_list, 11
trans, 11
coap_server.c
COAP_SERVER_ACK_TIMEOUT_SEC, 45
COAP_SERVER_MAX_RETRANSMIT, 45
coap_server_add_sep_resp_uri_path, 45
coap_server_create, 45
coap_server_destroy, 46
coap_server_get_next_msg_id, 46
coap_server_run, 46
coap_server.h
COAP_SERVER_ADDR_BUF_LEN, 35
COAP_SERVER_NUM_TRANS, 35
coap_server_add_sep_resp_uri_path, 35
coap_server_create, 35
coap_server_destroy, 36
coap_server_get_next_msg_id, 36
coap_server_run, 36
coap_server_add_sep_resp_uri_path
coap_server.c, 45
coap_server.h, 35
coap_server_create
coap_server.c, 45
coap_server.h, 35
coap_server_destroy
coap_server.c, 46
coap_server.h, 36
coap_server_get_next_msg_id
coap_server.c, 46
coap_server.h, 36
coap_server_path, 12
next, 12
str, 12
coap_server_path_list_t, 12
first, 12
last, 12
coap_server_run
coap_server.c, 46
coap_server.h, 36
coap_server_trans, 13

active, 13
client_addr, 13
client_sin, 13
client_sin_len, 13
last_use, 13
num_retrans, 14
req, 14
resp, 14
server, 14
timeout, 14
timer_fd, 14
code_class
coap_msg_t, 10
test_client_data_t, 15
test_msg_data_t, 17
code_detail
coap_msg_t, 10
test_client_data_t, 15
test_msg_data_t, 17
copy_desc
test_msg_data_t, 17
copy_ret
test_msg_data_t, 17
crl_file_name
test_client_data_t, 15

data
test_t, 20
desc
test_client_data_t, 15

FAIL_STR
test.c, 47
first
coap_msg_op_list_t, 9
coap_server_path_list_t, 12
format_desc
test_msg_data_t, 17
format_ret
test_msg_data_t, 17
func
test_t, 20

handle
coap_server, 11
host
test_client_data_t, 15

include/coap_client.h, 21
include/coap_log.h, 22
include/coap_msg.h, 25
include/coap_server.h, 34

key_file_name
test_client_data_t, 15

last
coap_msg_op_list_t, 9
coap_server_path_list_t, 12
last_use
coap_server_trans, 13
len
coap_msg_op, 8
test_msg_op_t, 19

msg_id
coap_msg_t, 10
coap_server, 11
test_msg_data_t, 17

next
coap_msg_op, 8
coap_server_path, 12
num
coap_msg_op, 8
test_msg_op_t, 19
num_ops
test_msg_data_t, 17
num_retrans
coap_client_t, 7
coap_server_trans, 14

op_list
coap_msg_t, 10
ops
test_msg_data_t, 17

PASS_STR
test.c, 47
parse_desc
test_msg_data_t, 17
parse_ret
test_msg_data_t, 17
payload
coap_msg_t, 10
test_client_data_t, 15
test_msg_data_t, 18
payload_len
coap_msg_t, 10
test_client_data_t, 15
test_msg_data_t, 18
port
test_client_data_t, 15

req
coap_server_trans, 14
resp
coap_server_trans, 14

START_STR
test.c, 47
sd
coap_client_t, 7
coap_server, 11
sep_list
coap_server, 11
server
coap_server_trans, 14
server_addr
coap_client_t, 7

server_sin
    coap_client_t, 7
server_sin_len
    coap_client_t, 7
set_code_ret
    test_msg_data_t, 18
set_msg_id_ret
    test_msg_data_t, 18
set_payload_ret
    test_msg_data_t, 18
set_token_ret
    test_msg_data_t, 18
set_type_ret
    test_msg_data_t, 18
src/coap_client.c, 37
src/coap_msg.c, 38
src/coap_server.c, 44
str
    coap_server_path, 12

test.c
    FAIL_STR, 47
    PASS_STR, 47
    START_STR, 47
    test_run, 47
test.h
    test_run, 48
test/test.c, 47
test/test.h, 48
test_client_data_t, 14
    cert_file_name, 15
    code_class, 15
    code_detail, 15
    crl_file_name, 15
    desc, 15
    host, 15
    key_file_name, 15
    payload, 15
    payload_len, 15
    port, 15
    trust_file_name, 15
    type, 15
    uri_path_opt, 16
test_msg_data_t, 16
    add_op_ret, 16
    buf, 16
    buf_len, 17
    code_class, 17
    code_detail, 17
    copy_desc, 17
    copy_ret, 17
    format_desc, 17
    format_ret, 17
    msg_id, 17
    num_ops, 17
    ops, 17
    parse_desc, 17
    parse_ret, 17
    payload, 18
    payload_len, 18
    set_code_ret, 18
    set_msg_id_ret, 18
    set_payload_ret, 18
    set_token_ret, 18
    set_type_ret, 18
    token, 18
    token_len, 18
    type, 18
    ver, 18
test_msg_op_t, 19
    len, 19
    num, 19
    val, 19
test_run
    test.c, 47
    test.h, 48
test_t, 19
    data, 20
    func, 20
timeout
    coap_client_t, 8
    coap_server_trans, 14
timer_fd
    coap_client_t, 8
    coap_server_trans, 14
token
    coap_msg_t, 10
    test_msg_data_t, 18
token_len
    coap_msg_t, 10
    test_msg_data_t, 18
trans
    coap_server, 11
trust_file_name
    test_client_data_t, 15
type
    coap_msg_t, 10
    test_client_data_t, 15
    test_msg_data_t, 18

uri_path_opt
    test_client_data_t, 16

val
    coap_msg_op, 8
    test_msg_op_t, 19
ver
    coap_msg_t, 10
    test_msg_data_t, 18