

FreeCoAP

Generated by Doxygen 1.8.9.1

Thu Sep 22 2016 18:00:53



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	coap_client_t Struct Reference . . . . .	5
3.1.1	Detailed Description . . . . .	5
3.1.2	Member Data Documentation . . . . .	5
3.1.2.1	app_len . . . . .	5
3.1.2.2	app_start . . . . .	5
3.1.2.3	ctx . . . . .	6
3.1.2.4	ecdsa_key . . . . .	6
3.1.2.5	num_retrans . . . . .	6
3.1.2.6	sd . . . . .	6
3.1.2.7	server_host . . . . .	6
3.1.2.8	server_port . . . . .	6
3.1.2.9	server_sin . . . . .	6
3.1.2.10	server_sin_len . . . . .	6
3.1.2.11	sess . . . . .	6
3.1.2.12	state . . . . .	6
3.1.2.13	timeout . . . . .	6
3.1.2.14	timer_fd . . . . .	6
3.2	coap_msg_op Struct Reference . . . . .	7
3.2.1	Detailed Description . . . . .	7
3.2.2	Member Data Documentation . . . . .	7
3.2.2.1	len . . . . .	7
3.2.2.2	next . . . . .	7
3.2.2.3	num . . . . .	7
3.2.2.4	val . . . . .	7
3.3	coap_msg_op_list_t Struct Reference . . . . .	8

3.3.1	Detailed Description . . . . .	8
3.3.2	Member Data Documentation . . . . .	8
3.3.2.1	first . . . . .	8
3.3.2.2	last . . . . .	8
3.4	coap_msg_t Struct Reference . . . . .	8
3.4.1	Detailed Description . . . . .	9
3.4.2	Member Data Documentation . . . . .	9
3.4.2.1	code_class . . . . .	9
3.4.2.2	code_detail . . . . .	9
3.4.2.3	msg_id . . . . .	9
3.4.2.4	op_list . . . . .	10
3.4.2.5	payload . . . . .	10
3.4.2.6	payload_len . . . . .	10
3.4.2.7	token . . . . .	10
3.4.2.8	token_len . . . . .	10
3.4.2.9	type . . . . .	10
3.4.2.10	ver . . . . .	10
3.5	coap_server Struct Reference . . . . .	10
3.5.1	Detailed Description . . . . .	11
3.5.2	Member Data Documentation . . . . .	11
3.5.2.1	ecdsa_key . . . . .	11
3.5.2.2	handle . . . . .	11
3.5.2.3	msg_id . . . . .	12
3.5.2.4	sd . . . . .	12
3.5.2.5	sep_list . . . . .	12
3.5.2.6	trans . . . . .	12
3.6	coap_server_path Struct Reference . . . . .	12
3.6.1	Detailed Description . . . . .	12
3.6.2	Member Data Documentation . . . . .	12
3.6.2.1	next . . . . .	12
3.6.2.2	str . . . . .	13
3.7	coap_server_path_list_t Struct Reference . . . . .	13
3.7.1	Detailed Description . . . . .	13
3.7.2	Member Data Documentation . . . . .	13
3.7.2.1	first . . . . .	13
3.7.2.2	last . . . . .	13
3.8	coap_server_trans Struct Reference . . . . .	14
3.8.1	Detailed Description . . . . .	15
3.8.2	Member Data Documentation . . . . .	15
3.8.2.1	active . . . . .	15

3.8.2.2	<a href="#">app_len</a>	15
3.8.2.3	<a href="#">app_start</a>	15
3.8.2.4	<a href="#">client_addr</a>	15
3.8.2.5	<a href="#">client_sin</a>	15
3.8.2.6	<a href="#">client_sin_len</a>	15
3.8.2.7	<a href="#">ctx</a>	15
3.8.2.8	<a href="#">last_use</a>	15
3.8.2.9	<a href="#">num_retrans</a>	15
3.8.2.10	<a href="#">req</a>	15
3.8.2.11	<a href="#">resp</a>	16
3.8.2.12	<a href="#">server</a>	16
3.8.2.13	<a href="#">sess</a>	16
3.8.2.14	<a href="#">state</a>	16
3.8.2.15	<a href="#">timeout</a>	16
3.8.2.16	<a href="#">timer_fd</a>	16
<b>4</b>	<b>File Documentation</b>	<b>17</b>
4.1	<a href="#">lib/include/coap_client.h File Reference</a>	17
4.1.1	<a href="#">Detailed Description</a>	18
4.1.2	<a href="#">Macro Definition Documentation</a>	18
4.1.2.1	<a href="#">coap_client_dtls_get_state</a>	18
4.1.2.2	<a href="#">COAP_CLIENT_HOST_BUF_LEN</a>	19
4.1.2.3	<a href="#">COAP_CLIENT_PORT_BUF_LEN</a>	19
4.1.3	<a href="#">Enumeration Type Documentation</a>	19
4.1.3.1	<a href="#">coap_client_dtls_state_t</a>	19
4.1.4	<a href="#">Function Documentation</a>	19
4.1.4.1	<a href="#">coap_client_create</a>	19
4.1.4.2	<a href="#">coap_client_destroy</a>	19
4.1.4.3	<a href="#">coap_client_exchange</a>	20
4.2	<a href="#">lib/include/coap_log.h File Reference</a>	20
4.2.1	<a href="#">Detailed Description</a>	21
4.2.2	<a href="#">Macro Definition Documentation</a>	21
4.2.2.1	<a href="#">COAP_LOG_DEF_LEVEL</a>	21
4.2.3	<a href="#">Enumeration Type Documentation</a>	21
4.2.3.1	<a href="#">coap_log_level_t</a>	21
4.2.4	<a href="#">Function Documentation</a>	21
4.2.4.1	<a href="#">coap_log_debug</a>	21
4.2.4.2	<a href="#">coap_log_error</a>	22
4.2.4.3	<a href="#">coap_log_get_level</a>	22
4.2.4.4	<a href="#">coap_log_info</a>	22

4.2.4.5	<a href="#">coap_log_notice</a>	22
4.2.4.6	<a href="#">coap_log_set_level</a>	22
4.2.4.7	<a href="#">coap_log_warn</a>	23
4.3	<a href="#">lib/include/coap_msg.h File Reference</a>	24
4.3.1	<a href="#">Detailed Description</a>	27
4.3.2	<a href="#">Macro Definition Documentation</a>	27
4.3.2.1	<a href="#">coap_msg_get_code_class</a>	27
4.3.2.2	<a href="#">coap_msg_get_code_detail</a>	27
4.3.2.3	<a href="#">coap_msg_get_first_op</a>	27
4.3.2.4	<a href="#">coap_msg_get_msg_id</a>	27
4.3.2.5	<a href="#">coap_msg_get_payload</a>	27
4.3.2.6	<a href="#">coap_msg_get_payload_len</a>	27
4.3.2.7	<a href="#">coap_msg_get_token</a>	27
4.3.2.8	<a href="#">coap_msg_get_token_len</a>	27
4.3.2.9	<a href="#">coap_msg_get_type</a>	27
4.3.2.10	<a href="#">coap_msg_get_ver</a>	28
4.3.2.11	<a href="#">coap_msg_is_empty</a>	28
4.3.2.12	<a href="#">COAP_MSG_MAX_BUF_LEN</a>	28
4.3.2.13	<a href="#">COAP_MSG_MAX_CODE_CLASS</a>	28
4.3.2.14	<a href="#">COAP_MSG_MAX_CODE_DETAIL</a>	28
4.3.2.15	<a href="#">COAP_MSG_MAX_MSG_ID</a>	28
4.3.2.16	<a href="#">COAP_MSG_MAX_TOKEN_LEN</a>	28
4.3.2.17	<a href="#">coap_msg_op_get_len</a>	28
4.3.2.18	<a href="#">coap_msg_op_get_next</a>	28
4.3.2.19	<a href="#">coap_msg_op_get_num</a>	28
4.3.2.20	<a href="#">coap_msg_op_get_val</a>	28
4.3.2.21	<a href="#">coap_msg_op_num_is_critical</a>	28
4.3.2.22	<a href="#">coap_msg_op_num_is_unsafe</a>	29
4.3.2.23	<a href="#">coap_msg_op_num_no_cache_key</a>	29
4.3.2.24	<a href="#">coap_msg_op_set_len</a>	29
4.3.2.25	<a href="#">coap_msg_op_set_next</a>	29
4.3.2.26	<a href="#">coap_msg_op_set_num</a>	29
4.3.2.27	<a href="#">coap_msg_op_set_val</a>	29
4.3.2.28	<a href="#">COAP_MSG_OP_URI_PATH_MAX_LEN</a>	29
4.3.2.29	<a href="#">COAP_MSG_OP_URI_PATH_NUM</a>	29
4.3.2.30	<a href="#">COAP_MSG_VER</a>	29
4.3.3	<a href="#">Enumeration Type Documentation</a>	29
4.3.3.1	<a href="#">coap_msg_class_t</a>	29
4.3.3.2	<a href="#">coap_msg_client_err_t</a>	30
4.3.3.3	<a href="#">coap_msg_method_t</a>	30

4.3.3.4	<a href="#">coap_msg_op_num_t</a>	30
4.3.3.5	<a href="#">coap_msg_server_err_t</a>	31
4.3.3.6	<a href="#">coap_msg_success_t</a>	31
4.3.3.7	<a href="#">coap_msg_type_t</a>	31
4.3.4	Function Documentation	31
4.3.4.1	<a href="#">coap_msg_add_op</a>	31
4.3.4.2	<a href="#">coap_msg_check_critical_ops</a>	32
4.3.4.3	<a href="#">coap_msg_check_unsafe_ops</a>	32
4.3.4.4	<a href="#">coap_msg_copy</a>	32
4.3.4.5	<a href="#">coap_msg_create</a>	33
4.3.4.6	<a href="#">coap_msg_destroy</a>	33
4.3.4.7	<a href="#">coap_msg_format</a>	33
4.3.4.8	<a href="#">coap_msg_gen_rand_str</a>	33
4.3.4.9	<a href="#">coap_msg_op_num_is_recognized</a>	34
4.3.4.10	<a href="#">coap_msg_parse</a>	34
4.3.4.11	<a href="#">coap_msg_parse_type_msg_id</a>	34
4.3.4.12	<a href="#">coap_msg_reset</a>	35
4.3.4.13	<a href="#">coap_msg_set_code</a>	35
4.3.4.14	<a href="#">coap_msg_set_msg_id</a>	35
4.3.4.15	<a href="#">coap_msg_set_payload</a>	35
4.3.4.16	<a href="#">coap_msg_set_token</a>	36
4.3.4.17	<a href="#">coap_msg_set_type</a>	36
4.4	<a href="#">lib/include/coap_server.h</a> File Reference	36
4.4.1	Detailed Description	38
4.4.2	Macro Definition Documentation	38
4.4.2.1	<a href="#">COAP_SERVER_ADDR_BUF_LEN</a>	38
4.4.2.2	<a href="#">COAP_SERVER_DIAG_PAYLOAD_LEN</a>	38
4.4.2.3	<a href="#">COAP_SERVER_NUM_TRANS</a>	38
4.4.3	Enumeration Type Documentation	39
4.4.3.1	<a href="#">coap_server_dtls_state_t</a>	39
4.4.3.2	<a href="#">coap_server_resp_t</a>	39
4.4.4	Function Documentation	39
4.4.4.1	<a href="#">coap_server_add_sep_resp_uri_path</a>	39
4.4.4.2	<a href="#">coap_server_create</a>	39
4.4.4.3	<a href="#">coap_server_destroy</a>	40
4.4.4.4	<a href="#">coap_server_get_next_msg_id</a>	40
4.4.4.5	<a href="#">coap_server_run</a>	40
4.5	<a href="#">lib/src/coap_client.c</a> File Reference	41
4.5.1	Detailed Description	42
4.5.2	Macro Definition Documentation	42

4.5.2.1	COAP_CLIENT_ACK_TIMEOUT_SEC	42
4.5.2.2	COAP_CLIENT_DTLS_RETRANS_TIMEOUT	42
4.5.2.3	COAP_CLIENT_DTLS_TOTAL_TIMEOUT	42
4.5.2.4	COAP_CLIENT_MAX_RETRANSMIT	42
4.5.2.5	COAP_CLIENT_RESP_TIMEOUT_SEC	42
4.5.3	Function Documentation	42
4.5.3.1	coap_client_create	42
4.5.3.2	coap_client_destroy	43
4.5.3.3	coap_client_exchange	43
4.6	lib/src/coap_log.c File Reference	43
4.6.1	Detailed Description	44
4.6.2	Function Documentation	44
4.6.2.1	coap_log_debug	44
4.6.2.2	coap_log_error	44
4.6.2.3	coap_log_get_level	44
4.6.2.4	coap_log_info	44
4.6.2.5	coap_log_notice	45
4.6.2.6	coap_log_set_level	45
4.6.2.7	coap_log_warn	45
4.7	lib/src/coap_msg.c File Reference	45
4.7.1	Detailed Description	47
4.7.2	Macro Definition Documentation	47
4.7.2.1	coap_msg_op_list_get_first	47
4.7.2.2	coap_msg_op_list_get_last	47
4.7.2.3	coap_msg_op_list_is_empty	47
4.7.3	Function Documentation	47
4.7.3.1	coap_msg_add_op	47
4.7.3.2	coap_msg_check_critical_ops	47
4.7.3.3	coap_msg_check_unsafe_ops	48
4.7.3.4	coap_msg_copy	48
4.7.3.5	coap_msg_create	48
4.7.3.6	coap_msg_destroy	49
4.7.3.7	coap_msg_format	49
4.7.3.8	coap_msg_gen_rand_str	49
4.7.3.9	coap_msg_op_num_is_recognized	49
4.7.3.10	coap_msg_parse	50
4.7.3.11	coap_msg_parse_type_msg_id	50
4.7.3.12	coap_msg_reset	50
4.7.3.13	coap_msg_set_code	51
4.7.3.14	coap_msg_set_msg_id	51



4.7.3.15	coap_msg_set_payload	51
4.7.3.16	coap_msg_set_token	52
4.7.3.17	coap_msg_set_type	52
4.8	lib/src/coap_server.c File Reference	52
4.8.1	Detailed Description	54
4.8.2	Macro Definition Documentation	54
4.8.2.1	COAP_SERVER_ACK_TIMEOUT_SEC	54
4.8.2.2	COAP_SERVER_DTLS_MTU	54
4.8.2.3	COAP_SERVER_DTLS_NUM_DH_BITS	54
4.8.2.4	COAP_SERVER_DTLS_PRIORITIES	54
4.8.2.5	COAP_SERVER_DTLS_RETRANS_TIMEOUT	54
4.8.2.6	COAP_SERVER_DTLS_TOTAL_TIMEOUT	54
4.8.2.7	COAP_SERVER_MAX_RETRANSMIT	54
4.8.3	Function Documentation	54
4.8.3.1	coap_server_add_sep_resp_uri_path	54
4.8.3.2	coap_server_create	55
4.8.3.3	coap_server_destroy	55
4.8.3.4	coap_server_get_next_msg_id	55
4.8.3.5	coap_server_run	55
<b>Index</b>		<b>57</b>



# Chapter 1

## Class Index

### 1.1 Class List

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

<a href="#">coap_client_t</a>		
Client structure	. . . . .	5
<a href="#">coap_msg_op</a>		
Option structure	. . . . .	7
<a href="#">coap_msg_op_list_t</a>		
Option linked-list structure	. . . . .	8
<a href="#">coap_msg_t</a>		
Message structure	. . . . .	8
<a href="#">coap_server</a>		
Server structure	. . . . .	10
<a href="#">coap_server_path</a>		
URI path structure	. . . . .	12
<a href="#">coap_server_path_list_t</a>		
URI path list structure	. . . . .	13
<a href="#">coap_server_trans</a>		
Transaction structure	. . . . .	14



## Chapter 2

# File Index

### 2.1 File List

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

lib/include/ <a href="#">coap_client.h</a>	
Include file for the FreeCoAP client library . . . . .	17
lib/include/ <a href="#">coap_log.h</a>	
Include file for the FreeCoAP logging module . . . . .	20
lib/include/ <a href="#">coap_msg.h</a>	
Include file for the FreeCoAP message parser/formatter library . . . . .	24
lib/include/ <a href="#">coap_server.h</a>	
Include file for the FreeCoAP server library . . . . .	36
lib/src/ <a href="#">coap_client.c</a>	
Source file for the FreeCoAP client library . . . . .	41
lib/src/ <a href="#">coap_log.c</a>	
Source file for the FreeCoAP logging module . . . . .	43
lib/src/ <a href="#">coap_msg.c</a>	
Source file for the FreeCoAP message parser/formatter library . . . . .	45
lib/src/ <a href="#">coap_server.c</a>	
Source file for the FreeCoAP server library . . . . .	52



## Chapter 3

# Class Documentation

### 3.1 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\\_host](#) [COAP\_CLIENT\_HOST\_BUF\_LEN]
- char [server\\_port](#) [COAP\_CLIENT\_PORT\_BUF\_LEN]
- [coap\\_client\\_dtls\\_state\\_t](#) [state](#)
- dtls\_context\_t \* [ctx](#)
- session\_t [sess](#)
- dtls\_ecdsa\_key\_t [ecdsa\\_key](#)
- char \* [app\\_start](#)
- size\_t [app\\_len](#)

#### 3.1.1 Detailed Description

Client structure.

#### 3.1.2 Member Data Documentation

##### 3.1.2.1 size\_t coap\_client\_t::app\_len

Length of application data

##### 3.1.2.2 char\* coap\_client\_t::app\_start

Start of application data

### 3.1.2.3 `dtls_context_t*` `coap_client_t::ctx`

DTLS context

### 3.1.2.4 `dtls_ecdsa_key_t` `coap_client_t::ecdsa_key`

ECDSA keys

### 3.1.2.5 `unsigned` `coap_client_t::num_retrans`

Current number of retransmissions

### 3.1.2.6 `int` `coap_client_t::sd`

Socket descriptor

### 3.1.2.7 `char` `coap_client_t::server_host[COAP_CLIENT_HOST_BUF_LEN]`

String to hold the server host address

### 3.1.2.8 `char` `coap_client_t::server_port[COAP_CLIENT_PORT_BUF_LEN]`

String to hold the server port number

### 3.1.2.9 `struct sockaddr_in6` `coap_client_t::server_sin`

IPv6 socket structure

### 3.1.2.10 `socklen_t` `coap_client_t::server_sin_len`

IPv6 socket structure length

### 3.1.2.11 `session_t` `coap_client_t::sess`

DTLS session

### 3.1.2.12 `coap_client_dtls_state_t` `coap_client_t::state`

Current state of the DTLS session

### 3.1.2.13 `struct timespec` `coap_client_t::timeout`

Timeout value

### 3.1.2.14 `int` `coap_client_t::timer_fd`

Timer file descriptor

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

- `lib/include/coap_client.h`

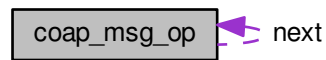


## 3.2 coap\_msg\_op Struct Reference

Option structure.

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

Collaboration diagram for coap\_msg\_op:



### Public Attributes

- unsigned [num](#)
- unsigned [len](#)
- char \* [val](#)
- struct [coap\\_msg\\_op](#) \* [next](#)

### 3.2.1 Detailed Description

Option structure.

### 3.2.2 Member Data Documentation

#### 3.2.2.1 unsigned coap\_msg\_op::len

Option length

#### 3.2.2.2 struct coap\_msg\_op\* coap\_msg\_op::next

Pointer to the next option structure in the list

#### 3.2.2.3 unsigned coap\_msg\_op::num

Option number

#### 3.2.2.4 char\* coap\_msg\_op::val

Pointer to a buffer containing the option value

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

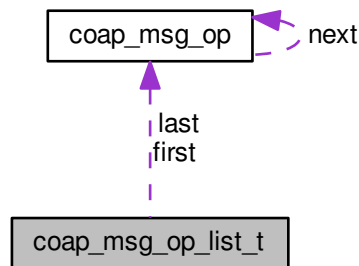
- lib/include/[coap\\_msg.h](#)

### 3.3 coap\_msg\_op\_list\_t Struct Reference

Option linked-list structure.

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

Collaboration diagram for coap\_msg\_op\_list\_t:



#### Public Attributes

- [coap\\_msg\\_op\\_t \\* first](#)
- [coap\\_msg\\_op\\_t \\* last](#)

#### 3.3.1 Detailed Description

Option linked-list structure.

#### 3.3.2 Member Data Documentation

##### 3.3.2.1 coap\_msg\_op\_t\* coap\_msg\_op\_list\_t::first

Pointer to the first option structure in the list

##### 3.3.2.2 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:

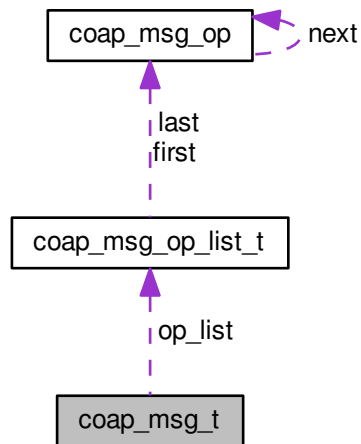
- lib/include/[coap\\_msg.h](#)

### 3.4 coap\_msg\_t Struct Reference

Message structure.

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

Collaboration diagram for coap\_msg\_t:



## 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`
- size\_t `payload_len`

### 3.4.1 Detailed Description

Message structure.

### 3.4.2 Member Data Documentation

#### 3.4.2.1 unsigned coap\_msg\_t::code\_class

Code class

#### 3.4.2.2 unsigned coap\_msg\_t::code\_detail

Code detail

#### 3.4.2.3 unsigned coap\_msg\_t::msg\_id

Message ID

#### 3.4.2.4 `coap_msg_op_list_t coap_msg_t::op_list`

Option list

#### 3.4.2.5 `char* coap_msg_t::payload`

Pointer to a buffer containing the payload

#### 3.4.2.6 `size_t coap_msg_t::payload_len`

Length of the payload

#### 3.4.2.7 `char coap_msg_t::token[COAP_MSG_MAX_TOKEN_LEN]`

Token value

#### 3.4.2.8 `unsigned coap_msg_t::token_len`

Token length

#### 3.4.2.9 `coap_msg_type_t coap_msg_t::type`

Message type

#### 3.4.2.10 `unsigned coap_msg_t::ver`

CoAP version

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

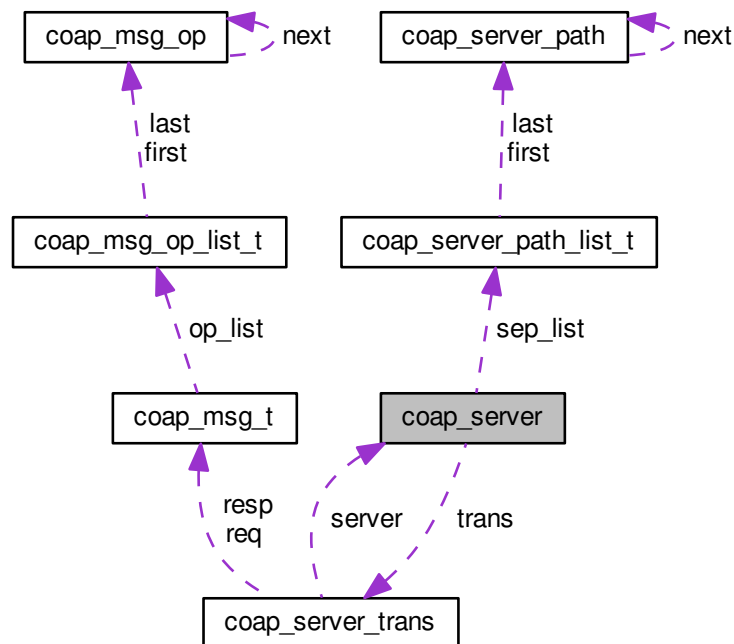
- [lib/include/coap\\_msg.h](#)

## 3.5 `coap_server` Struct Reference

Server structure.

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

Collaboration diagram for coap\_server:



## 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 *)`
- `dtls_ecdsa_key_t ecdsa_key`

### 3.5.1 Detailed Description

Server structure.

### 3.5.2 Member Data Documentation

#### 3.5.2.1 `dtls_ecdsa_key_t coap_server::ecdsa_key`

ECDSA keys

#### 3.5.2.2 `int(* coap_server::handle)(struct coap_server *, coap_msg_t *, coap_msg_t *)`

Call-back function to handle requests and generate responses

### 3.5.2.3 unsigned coap\_server::msg\_id

Last message ID value used in a response message

### 3.5.2.4 int coap\_server::sd

Socket descriptor

### 3.5.2.5 coap\_server\_path\_list\_t coap\_server::sep\_list

List of URI paths that require separate responses

### 3.5.2.6 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:

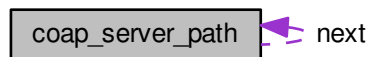
- [lib/include/coap\\_server.h](#)

## 3.6 coap\_server\_path Struct Reference

URI path structure.

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

Collaboration diagram for coap\_server\_path:



### Public Attributes

- char \* [str](#)
- struct [coap\\_server\\_path](#) \* [next](#)

### 3.6.1 Detailed Description

URI path structure.

### 3.6.2 Member Data Documentation

#### 3.6.2.1 struct coap\_server\_path\* coap\_server\_path::next

Pointer to the next URI path structure in the list

## 3.6.2.2 char\* coap\_server\_path::str

String containing a path

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

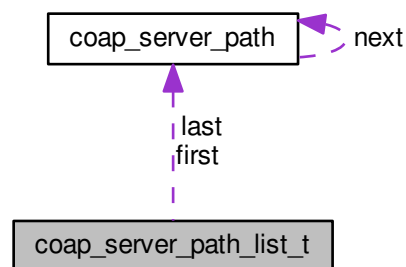
- lib/include/coap\_server.h

## 3.7 coap\_server\_path\_list\_t Struct Reference

URI path list structure.

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

Collaboration diagram for coap\_server\_path\_list\_t:



## Public Attributes

- [coap\\_server\\_path\\_t \\* first](#)
- [coap\\_server\\_path\\_t \\* last](#)

## 3.7.1 Detailed Description

URI path list structure.

## 3.7.2 Member Data Documentation

## 3.7.2.1 coap\_server\_path\_t\* coap\_server\_path\_list\_t::first

Pointer to the first URI path structure in the list

## 3.7.2.2 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:

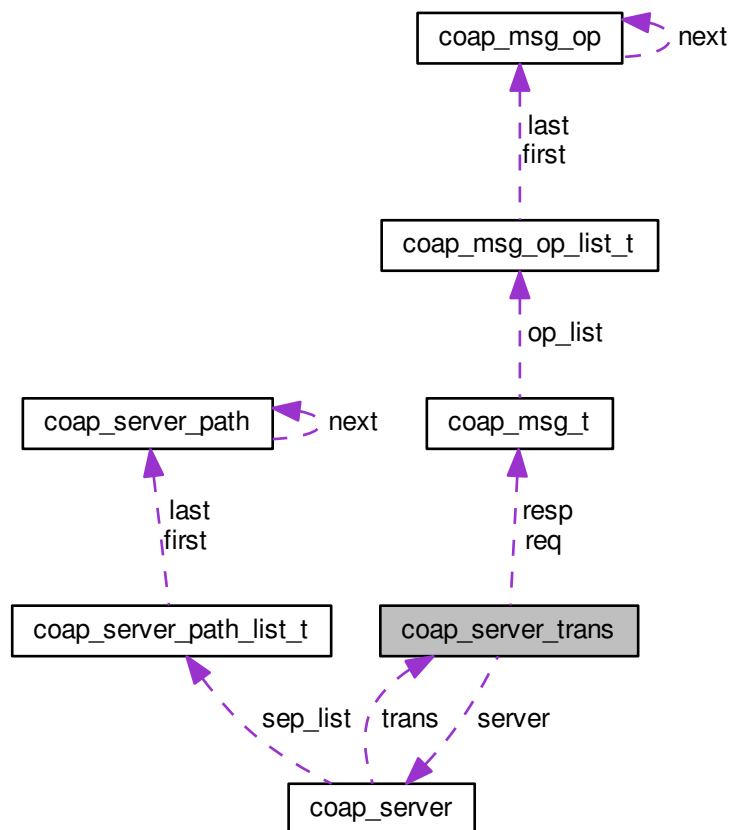
- lib/include/coap\_server.h

### 3.8 coap\_server\_trans Struct Reference

Transaction structure.

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

Collaboration diagram for coap\_server\_trans:



#### 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`
- coap\_server\_dtls\_state\_t `state`
- dtls\_context\_t \* `ctx`



- session\_t [sess](#)
- char \* [app\\_start](#)
- size\_t [app\\_len](#)

### 3.8.1 Detailed Description

Transaction structure.

### 3.8.2 Member Data Documentation

#### 3.8.2.1 int coap\_server\_trans::active

Flag to indicate if this transaction structure contains valid data

#### 3.8.2.2 size\_t coap\_server\_trans::app\_len

Length of application data

#### 3.8.2.3 char\* coap\_server\_trans::app\_start

Start of application data

#### 3.8.2.4 char coap\_server\_trans::client\_addr[COAP\_SERVER\_ADDR\_BUF\_LEN]

String to hold the client address

#### 3.8.2.5 struct sockaddr\_in6 coap\_server\_trans::client\_sin

IPv6 socket structure

#### 3.8.2.6 socklen\_t coap\_server\_trans::client\_sin\_len

IPv6 socket structure length

#### 3.8.2.7 dtls\_context\_t\* coap\_server\_trans::ctx

DTLS context

#### 3.8.2.8 time\_t coap\_server\_trans::last\_use

The time that this transaction structure was last used

#### 3.8.2.9 unsigned coap\_server\_trans::num\_retrans

Current number of retransmissions

#### 3.8.2.10 coap\_msg\_t coap\_server\_trans::req

Last request message received for this transaction

**3.8.2.11** `coap_msg_t coap_server_trans::resp`

Last response message sent for this transaction

**3.8.2.12** `struct coap_server* coap_server_trans::server`

Pointer to the containing server structure

**3.8.2.13** `session_t coap_server_trans::sess`

DTLS session

**3.8.2.14** `coap_server_dtls_state_t coap_server_trans::state`

Current state of the DTLS session

**3.8.2.15** `struct timespec coap_server_trans::timeout`

Timeout value

**3.8.2.16** `int coap_server_trans::timer_fd`

Timer file descriptor

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

- [lib/include/coap\\_server.h](#)

## Chapter 4

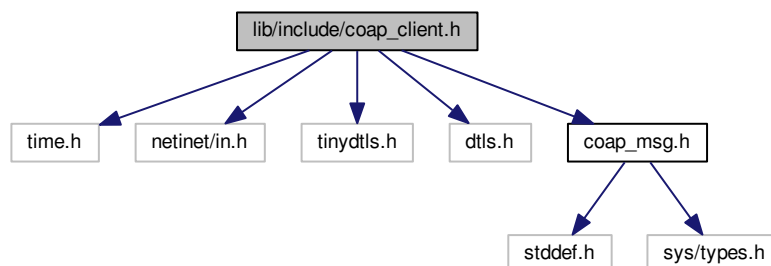
# File Documentation

### 4.1 lib/include/coap\_client.h File Reference

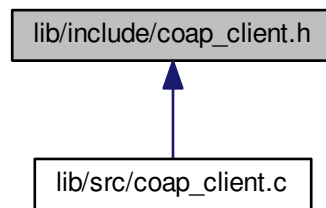
Include file for the FreeCoAP client library.

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

Include dependency graph for coap\_client.h:



This graph shows which files directly or indirectly include this file:



## Classes

- struct [coap\\_client\\_t](#)  
*Client structure.*

## Macros

- #define [COAP\\_CLIENT\\_HOST\\_BUF\\_LEN](#) 128
- #define [COAP\\_CLIENT\\_PORT\\_BUF\\_LEN](#) 8
- #define [coap\\_client\\_dtls\\_get\\_state](#)(client) ((client)->state)

## Enumerations

- enum [coap\\_client\\_dtls\\_state\\_t](#) { [COAP\\_CLIENT\\_DTLS\\_UNCONNECTED](#) = 0, [COAP\\_CLIENT\\_DTLS\\_CONNECTED](#), [COAP\\_CLIENT\\_DTLS\\_ALERT](#) }
- Client DTLS state enumeration.*

## Functions

- int [coap\\_client\\_create](#) ([coap\\_client\\_t](#) \*client, const char \*host, const char \*port, const unsigned char \*ecdsa\_priv\_key, const unsigned char \*ecdsa\_pub\_key\_x, const unsigned char \*ecdsa\_pub\_key\_y)  
*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.*

### 4.1.1 Detailed Description

Include file for the FreeCoAP client library.

### 4.1.2 Macro Definition Documentation

#### 4.1.2.1 #define coap\_client\_dtls\_get\_state( client )((client)->state)

Get DTLS state

## 4.1.2.2 #define COAP\_CLIENT\_HOST\_BUF\_LEN 128

Buffer length for host addresses

## 4.1.2.3 #define COAP\_CLIENT\_PORT\_BUF\_LEN 8

Buffer length for port numbers

## 4.1.3 Enumeration Type Documentation

## 4.1.3.1 enum coap\_client\_dtls\_state\_t

Client DTLS state enumeration.

Enumerator

**COAP\_CLIENT\_DTLS\_UNCONNECTED** DTLS session is not active

**COAP\_CLIENT\_DTLS\_CONNECTED** DTLS session is active

**COAP\_CLIENT\_DTLS\_ALERT** A DTLS alert message was received from the server

## 4.1.4 Function Documentation

## 4.1.4.1 int coap\_client\_create ( coap\_client\_t \* client, const char \* host, const char \* port, const unsigned char \* ecdsa\_priv\_key, const unsigned char \* ecdsa\_pub\_key\_x, const unsigned char \* ecdsa\_pub\_key\_y )

Initialise a client structure.

Parameters

out	<i>client</i>	Pointer to a client structure
in	<i>host</i>	Pointer to a string containing the host address of the server
in	<i>port</i>	Port number of the server
in	<i>ecdsa_priv_key</i>	Buffer containing the ECDSA private key
in	<i>ecdsa_pub_key_x</i>	Buffer containing the x component of the ECDSA public key
in	<i>ecdsa_pub_key_y</i>	Buffer containing the y component of the ECDSA public key

Returns

Operation status

Return values

0	Success
<0	Error

## 4.1.4.2 void coap\_client\_destroy ( coap\_client\_t \* client )

Deinitialise a client structure.

**Parameters**

<i>in, out</i>	<i>client</i>	Pointer to a client structure
----------------	---------------	-------------------------------

**4.1.4.3 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**

<i>in, out</i>	<i>client</i>	Pointer to a client structure
<i>in</i>	<i>req</i>	Pointer to the request message
<i>out</i>	<i>resp</i>	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

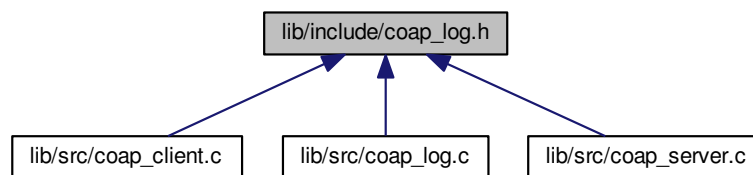
**Return values**

<i>0</i>	Success
<i>&lt;0</i>	Error

**4.2 lib/include/coap\_log.h File Reference**

Include file for the FreeCoAP logging module.

This graph shows which files directly or indirectly include this file:

**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.*

### 4.2.1 Detailed Description

Include file for the FreeCoAP logging module.

### 4.2.2 Macro Definition Documentation

#### 4.2.2.1 #define COAP\_LOG\_DEF\_LEVEL COAP\_LOG\_ERROR

Default log level

### 4.2.3 Enumeration Type Documentation

#### 4.2.3.1 enum coap\_log\_level\_t

Log level.

#### Enumerator

**COAP\_LOG\_ERROR** Error log level  
**COAP\_LOG\_WARN** Warning log level  
**COAP\_LOG\_NOTICE** Notice log level  
**COAP\_LOG\_INFO** Informational log level  
**COAP\_LOG\_DEBUG** Debug log level

### 4.2.4 Function Documentation

#### 4.2.4.1 void coap\_log\_debug ( const char \* msg, ... )

Log a debug message.

**Parameters**

in	<i>msg</i>	String containing format specifiers
in	...	arguments for the format specifiers

**4.2.4.2 void coap\_log\_error ( const char \* *msg*, ... )**

Log an error message.

**Parameters**

in	<i>msg</i>	String containing format specifiers
in	...	arguments for the format specifiers

**4.2.4.3 coap\_log\_level\_t coap\_log\_get\_level ( void )**

Get the log level.

**Returns**

The current log level

**4.2.4.4 void coap\_log\_info ( const char \* *msg*, ... )**

Log an info message.

**Parameters**

in	<i>msg</i>	String containing format specifiers
in	...	arguments for the format specifiers

**4.2.4.5 void coap\_log\_notice ( const char \* *msg*, ... )**

Log an notice message.

**Parameters**

in	<i>msg</i>	String containing format specifiers
in	...	arguments for the format specifiers

**4.2.4.6 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	<i>level</i>	The new log level
----	--------------	-------------------

< Warning log level

< Notice warning level

< Informational warning level

< Debug warning level



4.2.4.7 void coap\_log\_warn ( const char \* *msg*, ... )

Log a warning message.

## Parameters

in	<i>msg</i>	String containing format specifiers
in	...	arguments for the format specifiers

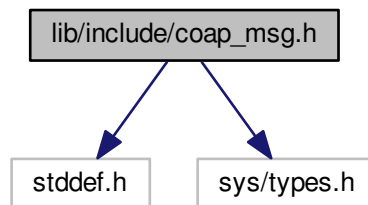
### 4.3 lib/include/coap\_msg.h File Reference

Include file for the FreeCoAP message parser/formatter library.

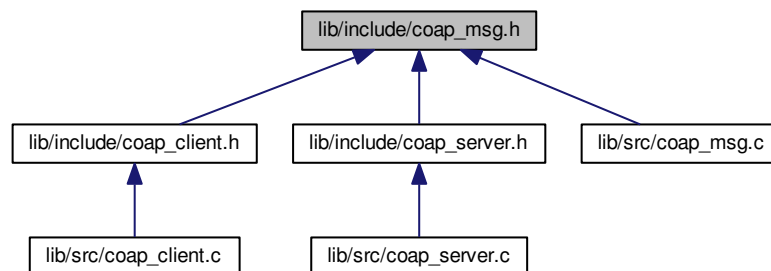
```
#include <stddef.h>
```

```
#include <sys/types.h>
```

Include dependency graph for coap\_msg.h:



This graph shows which files directly or indirectly include this file:



### 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, `COAP_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_CHANGED` = 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, `COAP_MSG_FORBIDDEN` = 3,  
`COAP_MSG_NOT_FOUND` = 4, `COAP_MSG_METHOD_NOT_ALLOWED` = 5, `COAP_MSG_NOT_ACCEPTABLE` = 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.*

- enum `coap_msg_op_num_t` {  
`COAP_MSG_IF_MATCH` = 1, `COAP_MSG_URI_HOST` = 3, `COAP_MSG_ETAG` = 4, `COAP_MSG_IF_NONE_MATCH` = 5,  
`COAP_MSG_URI_PORT` = 7, `COAP_MSG_LOCATION_PATH` = 8, `COAP_MSG_URI_PATH` = 11, `COAP_MSG_CONTENT_FORMAT` = 12,  
`COAP_MSG_MAX_AGE` = 14, `COAP_MSG_URI_QUERY` = 15, `COAP_MSG_ACCEPT` = 17, `COAP_MSG_LOCATION_QUERY` = 20,  
`COAP_MSG_PROXY_URI` = 35, `COAP_MSG_PROXY_SCHEME` = 39, `COAP_MSG_SIZE1` = 60 }

*Option number enumeration.*

## Functions

- int `coap_msg_op_num_is_recognized` (unsigned num)  
*Check if option is recognized.*
- void `coap_msg_gen_rand_str` (char \*buf, size\_t 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.*
- unsigned `coap_msg_check_critical_ops` (`coap_msg_t` \*msg)  
*Check that all of the critical options in a message are recognized.*
- unsigned `coap_msg_check_unsafe_ops` (`coap_msg_t` \*msg)  
*Check that all of the unsafe options in a message are recognized.*
- int `coap_msg_parse_type_msg_id` (char \*buf, size\_t len, unsigned \*type, unsigned \*msg\_id)  
*Extract the type and message ID values from a message.*
- ssize\_t `coap_msg_parse` (`coap_msg_t` \*msg, char \*buf, size\_t 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, size\_t 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, size\_t len)

*Set the payload in a message.*

- ssize\_t `coap_msg_format` (`coap_msg_t` \*msg, char \*buf, size\_t len)

*Format a message.*

- int `coap_msg_copy` (`coap_msg_t` \*dst, `coap_msg_t` \*src)

*Copy a message.*

### 4.3.1 Detailed Description

Include file for the FreeCoAP message parser/formatter library.

### 4.3.2 Macro Definition Documentation

4.3.2.1 `#define coap_msg_get_code_class( msg ) ((msg)->code_class)`

Get the code class from a message

4.3.2.2 `#define coap_msg_get_code_detail( msg ) ((msg)->code_detail)`

Get the code detail from a message

4.3.2.3 `#define coap_msg_get_first_op( msg ) ((msg)->op_list.first)`

Get the first option from a message

4.3.2.4 `#define coap_msg_get_msg_id( msg ) ((msg)->msg_id)`

Get the message ID from message

4.3.2.5 `#define coap_msg_get_payload( msg ) ((msg)->payload)`

Get the payload from a message

4.3.2.6 `#define coap_msg_get_payload_len( msg ) ((msg)->payload_len)`

Get the payload length from a message

4.3.2.7 `#define coap_msg_get_token( msg ) ((msg)->token)`

Get the token from a message

4.3.2.8 `#define coap_msg_get_token_len( msg ) ((msg)->token_len)`

Get the token length from a message

4.3.2.9 `#define coap_msg_get_type( msg ) ((msg)->type)`

Get the type from a message

4.3.2.10 `#define coap_msg_get_ver( msg ) ((msg)->ver)`

Get the version from a message

4.3.2.11 `#define coap_msg_is_empty( msg ) (((msg)->code_class == 0) && ((msg)->code_detail == 0))`

Indicate if a message is empty

4.3.2.12 `#define COAP_MSG_MAX_BUF_LEN 1152`

Maximum buffer length for header and payload

4.3.2.13 `#define COAP_MSG_MAX_CODE_CLASS 7`

Maximum code class

4.3.2.14 `#define COAP_MSG_MAX_CODE_DETAIL 31`

Maximum code detail

4.3.2.15 `#define COAP_MSG_MAX_MSG_ID ((1 << 16) - 1)`

Maximum message ID

4.3.2.16 `#define COAP_MSG_MAX_TOKEN_LEN 8`

Maximum token length

4.3.2.17 `#define coap_msg_op_get_len( op ) ((op)->len)`

Get the option length from an option

4.3.2.18 `#define coap_msg_op_get_next( op ) ((op)->next)`

Get the next pointer from an option

4.3.2.19 `#define coap_msg_op_get_num( op ) ((op)->num)`

Get the option number from an option

4.3.2.20 `#define coap_msg_op_get_val( op ) ((op)->val)`

Get the option value from an option

4.3.2.21 `#define coap_msg_op_num_is_critical( num ) ((num) & 1)`

Indicate if an option is critical

4.3.2.22 `#define coap_msg_op_num_is_unsafe( num ) ((num) & 2)`

Indicate if an option is unsafe to forward

4.3.2.23 `#define coap_msg_op_num_no_cache_key( num ) ((num & 0x1e) == 0x1c)`

Indicate if an option is not part of the cache key

4.3.2.24 `#define coap_msg_op_set_len( op, len ) ((op)->len = (len))`

Set the option length in an option

4.3.2.25 `#define coap_msg_op_set_next( op, next_op ) ((op)->next = (next_op))`

Set the next pointer in an option

4.3.2.26 `#define coap_msg_op_set_num( op, num ) ((op)->num = (num))`

Set the option number in an option

4.3.2.27 `#define coap_msg_op_set_val( op, val ) ((op)->val = (val))`

Set the option value in an option

4.3.2.28 `#define COAP_MSG_OP_URI_PATH_MAX_LEN 256`

Maximum buffer length for a reconstructed URI path

4.3.2.29 `#define COAP_MSG_OP_URI_PATH_NUM 11`

Uri-path option number

4.3.2.30 `#define COAP_MSG_VER 0x01`

CoAP version

### 4.3.3 Enumeration Type Documentation

4.3.3.1 `enum coap_msg_class_t`

Code class enumeration.

Enumerator

***COAP\_MSG\_REQ*** Request

***COAP\_MSG\_SUCCESS*** Success response

***COAP\_MSG\_CLIENT\_ERR*** Client error response

***COAP\_MSG\_SERVER\_ERR*** Server error response

#### 4.3.3.2 enum coap\_msg\_client\_err\_t

Client error response code detail enumeration.

Enumerator

**COAP\_MSG\_BAD\_REQ** Bad request client error  
**COAP\_MSG\_UNAUTHORIZED** Unauthorized client error  
**COAP\_MSG\_BAD\_OPTION** Bad option client error  
**COAP\_MSG\_FORBIDDEN** Forbidden client error  
**COAP\_MSG\_NOT\_FOUND** Not found client error  
**COAP\_MSG\_METHOD\_NOT\_ALLOWED** Method not allowed client error  
**COAP\_MSG\_NOT\_ACCEPTABLE** Not acceptable client error  
**COAP\_MSG\_PRECOND\_FAILED** Precondition failed client error  
**COAP\_MSG\_REQ\_ENT\_TOO\_LARGE** Request entity too large client error  
**COAP\_MSG\_UNSUP\_CONT\_FMT** Unsupported content-format client error

#### 4.3.3.3 enum coap\_msg\_method\_t

Code detail enumeration.

Enumerator

**COAP\_MSG\_GET** Get request method  
**COAP\_MSG\_POST** Post request method  
**COAP\_MSG\_PUT** Put request method  
**COAP\_MSG\_DELETE** Delete request method

#### 4.3.3.4 enum coap\_msg\_op\_num\_t

Option number enumeration.

Enumerator

**COAP\_MSG\_IF\_MATCH** If-Match option number  
**COAP\_MSG\_URI\_HOST** URI-Host option number  
**COAP\_MSG\_ETAG** Entity-Tag option number  
**COAP\_MSG\_IF\_NONE\_MATCH** If-None-Match option number  
**COAP\_MSG\_URI\_PORT** URI-Port option number  
**COAP\_MSG\_LOCATION\_PATH** Location-Path option number  
**COAP\_MSG\_URI\_PATH** URI-Path option number  
**COAP\_MSG\_CONTENT\_FORMAT** Content-Format option number  
**COAP\_MSG\_MAX\_AGE** Max-Age option number  
**COAP\_MSG\_URI\_QUERY** URI-Query option number  
**COAP\_MSG\_ACCEPT** Accept option number  
**COAP\_MSG\_LOCATION\_QUERY** Location-Query option number  
**COAP\_MSG\_PROXY\_URI** Proxy-URI option number  
**COAP\_MSG\_PROXY\_SCHEME** Proxy-Scheme option number  
**COAP\_MSG\_SIZE1** Size1 option number



## 4.3.3.5 enum coap\_msg\_server\_err\_t

Server error response code detail enumeration.

## Enumerator

**COAP\_MSG\_INT\_SERVER\_ERR** Internal server error  
**COAP\_MSG\_NOT\_IMPL** Not implemented server error  
**COAP\_MSG\_BAD\_GATEWAY** Bad gateway server error  
**COAP\_MSG\_SERV\_UNAVAIL** Service unavailable server error  
**COAP\_MSG\_GATEWAY\_TIMEOUT** Gateway timeout server error  
**COAP\_MSG\_PROXY\_NOT\_SUP** Proxying not supported server error

## 4.3.3.6 enum coap\_msg\_success\_t

Success response code detail enumeration.

## Enumerator

**COAP\_MSG\_CREATED** Created success response  
**COAP\_MSG\_DELETED** Deleted success response  
**COAP\_MSG\_VALID** Valid success response  
**COAP\_MSG\_CHANGED** Changed success response  
**COAP\_MSG\_CONTENT** Content success response

## 4.3.3.7 enum coap\_msg\_type\_t

Message type enumeration.

## Enumerator

**COAP\_MSG\_CON** Confirmable message  
**COAP\_MSG\_NON** Non-confirmable message  
**COAP\_MSG\_ACK** Acknowledgement message  
**COAP\_MSG\_RST** Reset message

## 4.3.4 Function Documentation

## 4.3.4.1 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	<i>msg</i>	Pointer to a message structure
in	<i>num</i>	Option number
in	<i>len</i>	Option length

<i>in</i>	<i>val</i>	Pointer to a buffer containing the option value
-----------	------------	---

**Returns**

Operation status

**Return values**

<i>0</i>	Success
<i>&lt;0</i>	Error

**4.3.4.2 unsigned coap\_msg\_check\_critical\_ops ( coap\_msg\_t \* msg )**

Check that all of the critical options in a message are recognized.

**Parameters**

<i>in</i>	<i>msg</i>	Pointer to message structure
-----------	------------	------------------------------

**Returns**

Operation status or bad option number

**Return values**

<i>0</i>	Success
<i>&gt;0</i>	Bad option number

**4.3.4.3 unsigned coap\_msg\_check\_unsafe\_ops ( coap\_msg\_t \* msg )**

Check that all of the unsafe options in a message are recognized.

**Parameters**

<i>in</i>	<i>msg</i>	Pointer to message structure
-----------	------------	------------------------------

**Returns**

Operation status or bad option number

**Return values**

<i>0</i>	Success
<i>&gt;0</i>	Bad option number

**4.3.4.4 int coap\_msg\_copy ( coap\_msg\_t \* dst, coap\_msg\_t \* src )**

Copy a message.

**Parameters**

<i>in, out</i>	<i>dst</i>	Pointer to the destination message structure
----------------	------------	--

<i>in</i>	<i>src</i>	Pointer to the source message structure
-----------	------------	---

**Returns**

Operation status

**Return values**

<i>0</i>	Success
<i>&lt;0</i>	Error

**4.3.4.5 void coap\_msg\_create ( coap\_msg\_t \* msg )**

Initialise a message structure.

**Parameters**

<i>out</i>	<i>msg</i>	Pointer to a message structure
------------	------------	--------------------------------

**4.3.4.6 void coap\_msg\_destroy ( coap\_msg\_t \* msg )**

Deinitialise a message structure.

**Parameters**

<i>in, out</i>	<i>msg</i>	Pointer to a message structure
----------------	------------	--------------------------------

**4.3.4.7 ssize\_t coap\_msg\_format ( coap\_msg\_t \* msg, char \* buf, size\_t len )**

Format a message.

**Parameters**

<i>in</i>	<i>msg</i>	Pointer to a message structure
<i>out</i>	<i>buf</i>	Pointer to a buffer to contain the formatted message
<i>in</i>	<i>len</i>	Length of the buffer

**Returns**

Length of the formatted message or error code

**Return values**

<i>&gt;0</i>	Length of the formatted message
<i>&lt;0</i>	Error

**4.3.4.8 void coap\_msg\_gen\_rand\_str ( char \* buf, size\_t len )**

Generate a random string of bytes.

**Parameters**

out	<i>buf</i>	Pointer to the buffer to store the random string
in	<i>len</i>	Length of the buffer

#### 4.3.4.9 int coap\_msg\_op\_num\_is\_recognized ( unsigned *num* )

Check if option is recognized.

##### Parameters

in	<i>num</i>	Option number
----	------------	---------------

##### Returns

Operation status

##### Return values

1	Option is recognized
0	Option is not recognized

#### 4.3.4.10 ssize\_t coap\_msg\_parse ( coap\_msg\_t \* *msg*, char \* *buf*, size\_t *len* )

Parse a message.

##### Parameters

in, out	<i>msg</i>	Pointer to a message structure
in	<i>buf</i>	Pointer to a buffer containing the message
in	<i>len</i>	Length of the buffer

##### Returns

Operation status

##### Return values

0	Success
<0	Error

#### 4.3.4.11 int coap\_msg\_parse\_type\_msg\_id ( char \* *buf*, size\_t *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	<i>buf</i>	Pointer to a buffer containing the message
in	<i>len</i>	Length of the buffer
out	<i>type</i>	Pointer to field to store the type value
out	<i>msg_id</i>	Pointer to a field to store the message ID value

##### Returns

Operation status

## Return values

0	Success
<0	Error

## 4.3.4.12 void coap\_msg\_reset ( coap\_msg\_t \* msg )

Deinitialise and initialise a message structure.

## Parameters

in, out	msg	Pointer to a message structure
---------	-----	--------------------------------

## 4.3.4.13 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

## 4.3.4.14 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

## 4.3.4.15 int coap\_msg\_set\_payload ( coap\_msg\_t \* msg, char \* buf, size\_t 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	<i>msg</i>	Pointer to a message structure
in	<i>buf</i>	Pointer to a buffer containing the payload
in	<i>len</i>	Length of the buffer

**Returns**

Operation status

**Return values**

0	Success
<0	Error

**4.3.4.16 int coap\_msg\_set\_token ( coap\_msg\_t \* msg, char \* buf, size\_t len )**

Set the token in a message.

**Parameters**

out	<i>msg</i>	Pointer to a message structure
in	<i>buf</i>	Pointer to a buffer containing the token
in	<i>len</i>	Length of the buffer

**Returns**

Operation status

**Return values**

0	Success
<0	Error

**4.3.4.17 int coap\_msg\_set\_type ( coap\_msg\_t \* msg, unsigned type )**

Set the type in a message.

**Parameters**

out	<i>msg</i>	Pointer to a message structure
in	<i>type</i>	Message type

**Returns**

Operation status

**Return values**

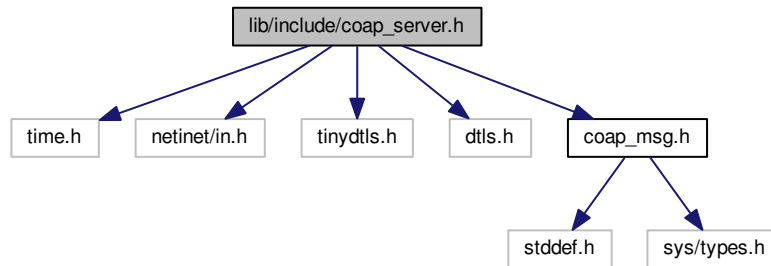
0	Success
<0	Error

**4.4 lib/include/coap\_server.h File Reference**

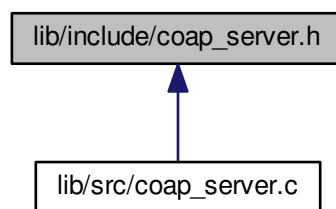
Include file for the FreeCoAP server library.

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

Include dependency graph for coap\_server.h:



This graph shows which files directly or indirectly include this file:



## 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`
- `#define COAP_SERVER_DIAG_PAYLOAD_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.*
- enum `coap_server_dtls_state_t` { `COAP_SERVER_DTLS_UNCONNECTED` = 0, `COAP_SERVER_DTLS_CONNECTED`, `COAP_SERVER_DTLS_ALERT` }  
*Server DTLS state enumeration.*

## Functions

- int `coap_server_create` (`coap_server_t` \*server, int(\*handle)(`coap_server_t` \*, `coap_msg_t` \*, `coap_msg_t` \*), const char \*host, const char \*port, const unsigned char \*ecdsa\_priv\_key, const unsigned char \*ecdsa\_pub\_key\_x, const unsigned char \*ecdsa\_pub\_key\_y)  
*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.*

### 4.4.1 Detailed Description

Include file for the FreeCoAP server library.

### 4.4.2 Macro Definition Documentation

#### 4.4.2.1 #define COAP\_SERVER\_ADDR\_BUF\_LEN 128

Buffer length for host addresses

#### 4.4.2.2 #define COAP\_SERVER\_DIAG\_PAYLOAD\_LEN 128

Buffer length for diagnostic payloads

#### 4.4.2.3 #define COAP\_SERVER\_NUM\_TRANS 8

Maximum number of active transactions per server



### 4.4.3 Enumeration Type Documentation

#### 4.4.3.1 enum coap\_server\_dtls\_state\_t

Server DTLS state enumeration.

Enumerator

- COAP\_SERVER\_DTLS\_UNCONNECTED** DTLS session is not active
- COAP\_SERVER\_DTLS\_CONNECTED** DTLS session is active
- COAP\_SERVER\_DTLS\_ALERT** A DTLS alert message was received from the client

#### 4.4.3.2 enum coap\_server\_resp\_t

Response type enumeration.

Enumerator

- COAP\_SERVER\_PIGGYBACKED** Piggybacked response
- COAP\_SERVER\_SEPARATE** Separate response

### 4.4.4 Function Documentation

#### 4.4.4.1 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	<i>server</i>	Pointer to a server structure
in	<i>str</i>	String representation of a URI path

Returns

Operation status

Return values

0	Success
<0	Error

#### 4.4.4.2 int coap\_server\_create ( coap\_server\_t \* server, int (\*)(coap\_server\_t \*, coap\_msg\_t \*, coap\_msg\_t \*) handle, const char \* host, const char \* port, const unsigned char \* ecdsa\_priv\_key, const unsigned char \* ecdsa\_pub\_key\_x, const unsigned char \* ecdsa\_pub\_key\_y )

Initialise a server structure.

Parameters

out	<i>server</i>	Pointer to a server structure
in	<i>handle</i>	Call-back function to handle client requests

in	<i>host</i>	Pointer to a string containing the host address of the server
in	<i>port</i>	Port number of the server
in	<i>ecdsa_priv_key</i>	Buffer containing the ECDSA private key
in	<i>ecdsa_pub_key_x</i>	Buffer containing the x component of the ECDSA public key
in	<i>ecdsa_pub_key_y</i>	Buffer containing the y component of the ECDSA public key

**Returns**

Operation status

**Return values**

0	Success
<0	Error

**4.4.4.3 void coap\_server\_destroy ( coap\_server\_t \* server )**

Deinitialise a server structure.

**Parameters**

in, out	<i>server</i>	Pointer to a server structure
---------	---------------	-------------------------------

**4.4.4.4 unsigned coap\_server\_get\_next\_msg\_id ( coap\_server\_t \* server )**

Get a new message ID value.

**Parameters**

in, out	<i>server</i>	Pointer to a server structure
---------	---------------	-------------------------------

**Returns**

message ID value

**4.4.4.5 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	<i>server</i>	Pointer to a server structure
---------	---------------	-------------------------------

**Returns**

Operation status

## Return values

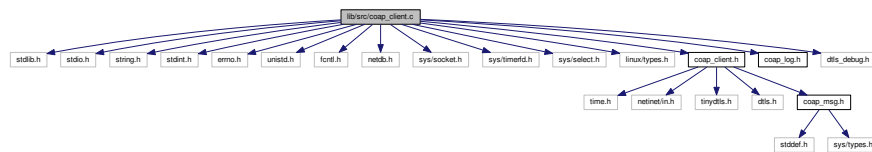
0	Success
<0	Error

## 4.5 lib/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 <netdb.h>
#include <sys/socket.h>
#include <sys/timerfd.h>
#include <sys/select.h>
#include <linux/types.h>
#include "coap_client.h"
#include "coap_log.h"
#include "dtls_debug.h"
```

Include dependency graph for coap\_client.c:



## Macros

- `#define COAP_CLIENT_ACK_TIMEOUT_SEC 2`
- `#define COAP_CLIENT_MAX_RETRANSMIT 4`
- `#define COAP_CLIENT_RESP_TIMEOUT_SEC 30`
- `#define COAP_CLIENT_DTLS_RETRANS_TIMEOUT 100`
- `#define COAP_CLIENT_DTLS_TOTAL_TIMEOUT 5000`

## Functions

- `int coap_client_create (coap_client_t *client, const char *host, const char *port, const unsigned char *ecdsa_priv_key, const unsigned char *ecdsa_pub_key_x, const unsigned char *ecdsa_pub_key_y)`  
*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.*

### 4.5.1 Detailed Description

Source file for the FreeCoAP client library.

### 4.5.2 Macro Definition Documentation

#### 4.5.2.1 `#define COAP_CLIENT_ACK_TIMEOUT_SEC 2`

Minimum delay to wait before retransmitting a confirmable message

#### 4.5.2.2 `#define COAP_CLIENT_DTLS_RETRANS_TIMEOUT 100`

Retransmission timeout (msec) for the DTLS handshake

#### 4.5.2.3 `#define COAP_CLIENT_DTLS_TOTAL_TIMEOUT 5000`

Total timeout (msec) for the DTLS handshake DTLS priorities

#### 4.5.2.4 `#define COAP_CLIENT_MAX_RETRANSMIT 4`

Maximum number of times a confirmable message can be retransmitted

#### 4.5.2.5 `#define COAP_CLIENT_RESP_TIMEOUT_SEC 30`

Maximum amount of time to wait for a response

### 4.5.3 Function Documentation

#### 4.5.3.1 `int coap_client_create ( coap_client_t * client, const char * host, const char * port, const unsigned char * ecdsa_priv_key, const unsigned char * ecdsa_pub_key_x, const unsigned char * ecdsa_pub_key_y )`

Initialise a client structure.

##### Parameters

out	<i>client</i>	Pointer to a client structure
in	<i>host</i>	Pointer to a string containing the host address of the server
in	<i>port</i>	Port number of the server
in	<i>ecdsa_priv_key</i>	Buffer containing the ECDSA private key
in	<i>ecdsa_pub_key_x</i>	Buffer containing the x component of the ECDSA public key
in	<i>ecdsa_pub_key_y</i>	Buffer containing the y component of the ECDSA public key

##### Returns

Operation status

##### Return values

0	Success
<0	Error

#### 4.5.3.2 void coap\_client\_destroy ( coap\_client\_t \* client )

Deinitialise a client structure.

##### Parameters

in, out	client	Pointer to a client structure
---------	--------	-------------------------------

#### 4.5.3.3 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

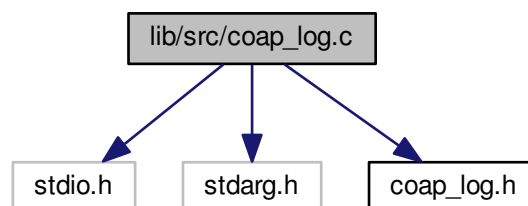
0	Success
<0	Error

## 4.6 lib/src/coap\_log.c File Reference

Source file for the FreeCoAP logging module.

```
#include <stdio.h>
#include <stdarg.h>
#include "coap_log.h"
```

Include dependency graph for coap\_log.c:



## 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.*

### 4.6.1 Detailed Description

Source file for the FreeCoAP logging module.

### 4.6.2 Function Documentation

#### 4.6.2.1 void [coap\\_log\\_debug](#) ( const char \* *msg*, ... )

Log a debug message.

##### Parameters

in	<i>msg</i>	String containing format specifiers
in	...	arguments for the format specifiers

#### 4.6.2.2 void [coap\\_log\\_error](#) ( const char \* *msg*, ... )

Log an error message.

##### Parameters

in	<i>msg</i>	String containing format specifiers
in	...	arguments for the format specifiers

#### 4.6.2.3 [coap\\_log\\_level\\_t](#) [coap\\_log\\_get\\_level](#) ( void )

Get the log level.

##### Returns

The current log level

#### 4.6.2.4 void [coap\\_log\\_info](#) ( const char \* *msg*, ... )

Log an info message.

## Parameters

in	<i>msg</i>	String containing format specifiers
in	...	arguments for the format specifiers

4.6.2.5 void coap\_log\_notice ( const char \* *msg*, ... )

Log an notice message.

## Parameters

in	<i>msg</i>	String containing format specifiers
in	...	arguments for the format specifiers

4.6.2.6 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	<i>level</i>	The new log level
----	--------------	-------------------

< Warning log level

< Notice warning level

< Informational warning level

< Debug warning level

4.6.2.7 void coap\_log\_warn ( const char \* *msg*, ... )

Log a warning message.

## Parameters

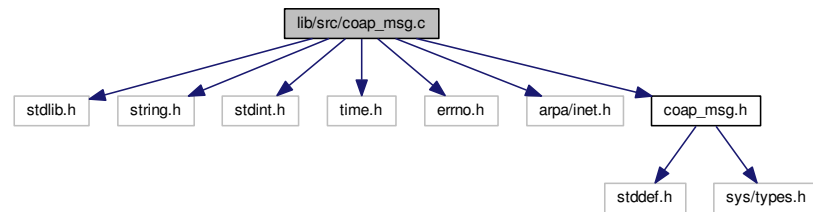
in	<i>msg</i>	String containing format specifiers
in	...	arguments for the format specifiers

## 4.7 lib/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"
```

Include dependency graph for `coap_msg.c`:



## 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, size\_t len)  
*Generate a random string of bytes.*
- int `coap_msg_op_num_is_recognized` (unsigned num)  
*Check if option is recognized.*
- 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.*
- unsigned `coap_msg_check_critical_ops` (coap\_msg\_t \*msg)  
*Check that all of the critical options in a message are recognized.*
- unsigned `coap_msg_check_unsafe_ops` (coap\_msg\_t \*msg)  
*Check that all of the unsafe options in a message are recognized.*
- int `coap_msg_parse_type_msg_id` (char \*buf, size\_t len, unsigned \*type, unsigned \*msg\_id)  
*Extract the type and message ID values from a message.*
- ssize\_t `coap_msg_parse` (coap\_msg\_t \*msg, char \*buf, size\_t 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, size\_t 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, size\_t len)  
*Set the payload in a message.*



- ssize\_t [coap\\_msg\\_format](#) ([coap\\_msg\\_t](#) \*msg, char \*buf, size\_t len)  
*Format a message.*
- int [coap\\_msg\\_copy](#) ([coap\\_msg\\_t](#) \*dst, [coap\\_msg\\_t](#) \*src)  
*Copy a message.*

### 4.7.1 Detailed Description

Source file for the FreeCoAP message parser/formatter library.

### 4.7.2 Macro Definition Documentation

#### 4.7.2.1 #define coap\_msg\_op\_list\_get\_first( list ) ((list)->first)

Get the first option from an option linked-list

#### 4.7.2.2 #define coap\_msg\_op\_list\_get\_last( list ) ((list)->last)

Get the last option in an option linked-list

#### 4.7.2.3 #define coap\_msg\_op\_list\_is\_empty( list ) ((list)->first == NULL)

Indicate whether or not an option linked-list is empty

### 4.7.3 Function Documentation

#### 4.7.3.1 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	<i>msg</i>	Pointer to a message structure
in	<i>num</i>	Option number
in	<i>len</i>	Option length
in	<i>val</i>	Pointer to a buffer containing the option value

##### Returns

Operation status

##### Return values

0	Success
<0	Error

#### 4.7.3.2 unsigned coap\_msg\_check\_critical\_ops ( coap\_msg\_t \* msg )

Check that all of the critical options in a message are recognized.

**Parameters**

<i>in</i>	<i>msg</i>	Pointer to message structure
-----------	------------	------------------------------

**Returns**

Operation status or bad option number

**Return values**

<i>0</i>	Success
<i>&gt;0</i>	Bad option number

**4.7.3.3 unsigned coap\_msg\_check\_unsafe\_ops ( coap\_msg\_t \* msg )**

Check that all of the unsafe options in a message are recognized.

**Parameters**

<i>in</i>	<i>msg</i>	Pointer to message structure
-----------	------------	------------------------------

**Returns**

Operation status or bad option number

**Return values**

<i>0</i>	Success
<i>&gt;0</i>	Bad option number

**4.7.3.4 int coap\_msg\_copy ( coap\_msg\_t \* dst, coap\_msg\_t \* src )**

Copy a message.

**Parameters**

<i>in, out</i>	<i>dst</i>	Pointer to the destination message structure
<i>in</i>	<i>src</i>	Pointer to the source message structure

**Returns**

Operation status

**Return values**

<i>0</i>	Success
<i>&lt;0</i>	Error

**4.7.3.5 void coap\_msg\_create ( coap\_msg\_t \* msg )**

Initialise a message structure.

## Parameters

out	<i>msg</i>	Pointer to a message structure
-----	------------	--------------------------------

4.7.3.6 void coap\_msg\_destroy ( coap\_msg\_t \* *msg* )

Deinitialise a message structure.

## Parameters

in, out	<i>msg</i>	Pointer to a message structure
---------	------------	--------------------------------

4.7.3.7 ssize\_t coap\_msg\_format ( coap\_msg\_t \* *msg*, char \* *buf*, size\_t *len* )

Format a message.

## Parameters

in	<i>msg</i>	Pointer to a message structure
out	<i>buf</i>	Pointer to a buffer to contain the formatted message
in	<i>len</i>	Length of the buffer

## Returns

Length of the formatted message or error code

## Return values

>0	Length of the formatted message
<0	Error

4.7.3.8 void coap\_msg\_gen\_rand\_str ( char \* *buf*, size\_t *len* )

Generate a random string of bytes.

## Parameters

out	<i>buf</i>	Pointer to the buffer to store the random string
in	<i>len</i>	Length of the buffer

4.7.3.9 int coap\_msg\_op\_num\_is\_recognized ( unsigned *num* )

Check if option is recognized.

## Parameters

in	<i>num</i>	Option number
----	------------	---------------

## Returns

Operation status

## Return values

1	Option is recognized
0	Option is not recognized

4.7.3.10 `ssize_t coap_msg_parse ( coap_msg_t * msg, char * buf, size_t len )`

Parse a message.

## Parameters

in, out	<i>msg</i>	Pointer to a message structure
in	<i>buf</i>	Pointer to a buffer containing the message
in	<i>len</i>	Length of the buffer

## Returns

Operation status

## Return values

0	Success
<0	Error

4.7.3.11 `int coap_msg_parse_type_msg_id ( char * buf, size_t 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	<i>buf</i>	Pointer to a buffer containing the message
in	<i>len</i>	Length of the buffer
out	<i>type</i>	Pointer to field to store the type value
out	<i>msg_id</i>	Pointer to a field to store the message ID value

## Returns

Operation status

## Return values

0	Success
<0	Error

4.7.3.12 `void coap_msg_reset ( coap_msg_t * msg )`

Deinitialise and initialise a message structure.

## Parameters

---

<i>in, out</i>	<i>msg</i>	Pointer to a message structure
----------------	------------	--------------------------------

#### 4.7.3.13 int coap\_msg\_set\_code ( coap\_msg\_t \* *msg*, unsigned *code\_class*, unsigned *code\_detail* )

Set the code in a message.

##### Parameters

<i>out</i>	<i>msg</i>	Pointer to a message structure
<i>in</i>	<i>code_class</i>	Code class
<i>in</i>	<i>code_detail</i>	Code detail

##### Returns

Operation status

##### Return values

<i>0</i>	Success
<i>&lt;0</i>	Error

#### 4.7.3.14 int coap\_msg\_set\_msg\_id ( coap\_msg\_t \* *msg*, unsigned *msg\_id* )

Set the message ID in a message.

##### Parameters

<i>out</i>	<i>msg</i>	Pointer to a message structure
<i>in</i>	<i>msg_id</i>	Message ID

##### Returns

Operation status

##### Return values

<i>0</i>	Success
<i>&lt;0</i>	Error

#### 4.7.3.15 int coap\_msg\_set\_payload ( coap\_msg\_t \* *msg*, char \* *buf*, size\_t *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

<i>in, out</i>	<i>msg</i>	Pointer to a message structure
<i>in</i>	<i>buf</i>	Pointer to a buffer containing the payload
<i>in</i>	<i>len</i>	Length of the buffer

##### Returns

Operation status

## Return values

<i>0</i>	Success
<i>&lt;0</i>	Error

4.7.3.16 `int coap_msg_set_token ( coap_msg_t * msg, char * buf, size_t len )`

Set the token in a message.

## Parameters

out	<i>msg</i>	Pointer to a message structure
in	<i>buf</i>	Pointer to a buffer containing the token
in	<i>len</i>	Length of the buffer

## Returns

Operation status

## Return values

<i>0</i>	Success
<i>&lt;0</i>	Error

4.7.3.17 `int coap_msg_set_type ( coap_msg_t * msg, unsigned type )`

Set the type in a message.

## Parameters

out	<i>msg</i>	Pointer to a message structure
in	<i>type</i>	Message type

## Returns

Operation status

## Return values

<i>0</i>	Success
<i>&lt;0</i>	Error

## 4.8 lib/src/coap\_server.c File Reference

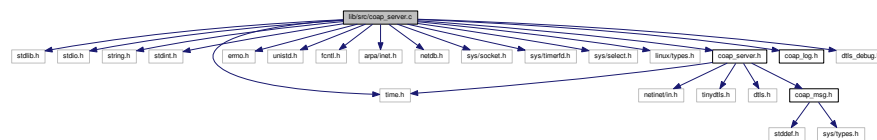
Source file for the FreeCoAP server library.

```

#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 <netdb.h>
#include <sys/socket.h>
#include <sys/timerfd.h>
#include <sys/select.h>
#include <linux/types.h>
#include "coap_server.h"
#include "coap_log.h"
#include "dtls_debug.h"

```

Include dependency graph for coap\_server.c:



## Macros

- #define [COAP\\_SERVER\\_ACK\\_TIMEOUT\\_SEC](#) 2
- #define [COAP\\_SERVER\\_MAX\\_RETRANSMIT](#) 4
- #define [COAP\\_SERVER\\_DTLS\\_MTU](#) [COAP\\_MSG\\_MAX\\_BUF\\_LEN](#)
- #define [COAP\\_SERVER\\_DTLS\\_RETRANS\\_TIMEOUT](#) 100
- #define [COAP\\_SERVER\\_DTLS\\_TOTAL\\_TIMEOUT](#) 5000
- #define [COAP\\_SERVER\\_DTLS\\_NUM\\_DH\\_BITS](#) 1024
- #define [COAP\\_SERVER\\_DTLS\\_PRIORITIES](#) "PERFORMANCE:-VERS-TLS-ALL:+VERS-DTLS1.0:%SE↵  
RVER\_PRECEDENCE"

## Functions

- int [coap\\_server\\_create](#) ([coap\\_server\\_t](#) \*server, int(\*handle)([coap\\_server\\_t](#) \*, [coap\\_msg\\_t](#) \*, [coap\\_msg\\_t](#) \*), const char \*host, const char \*port, const unsigned char \*ecdsa\_priv\_key, const unsigned char \*ecdsa↵  
\_pub\_key\_x, const unsigned char \*ecdsa\_pub\_key\_y)  
*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.*

### 4.8.1 Detailed Description

Source file for the FreeCoAP server library.

### 4.8.2 Macro Definition Documentation

#### 4.8.2.1 `#define COAP_SERVER_ACK_TIMEOUT_SEC 2`

Minimum delay to wait before retransmitting a confirmable message

#### 4.8.2.2 `#define COAP_SERVER_DTLS_MTU COAP_MSG_MAX_BUF_LEN`

Maximum transmission unit excluding the UDP and IPv6 headers

#### 4.8.2.3 `#define COAP_SERVER_DTLS_NUM_DH_BITS 1024`

DTLS Diffie-Hellman key size

#### 4.8.2.4 `#define COAP_SERVER_DTLS_PRIORITIES "PERFORMANCE:-VERS-TLS-ALL:+VERS-DTLS1.0:%SERVER_PRECEDE↵NCE"`

DTLS priorities

#### 4.8.2.5 `#define COAP_SERVER_DTLS_RETRANS_TIMEOUT 100`

Retransmission timeout (msec) for the DTLS handshake

#### 4.8.2.6 `#define COAP_SERVER_DTLS_TOTAL_TIMEOUT 5000`

Total timeout (msec) for the DTLS handshake

#### 4.8.2.7 `#define COAP_SERVER_MAX_RETRANSMIT 4`

Maximum number of times a confirmable message can be retransmitted

### 4.8.3 Function Documentation

#### 4.8.3.1 `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

<code>in, out</code>	<code>server</code>	Pointer to a server structure
<code>in</code>	<code>str</code>	String representation of a URI path

Returns

Operation status



## Return values

0	Success
<0	Error

**4.8.3.2** `int coap_server_create ( coap_server_t * server, int (*)(coap_server_t *, coap_msg_t *, coap_msg_t *) handle, const char * host, const char * port, const unsigned char * ecdsa_priv_key, const unsigned char * ecdsa_pub_key_x, const unsigned char * ecdsa_pub_key_y )`

Initialise a server structure.

## Parameters

out	<i>server</i>	Pointer to a server structure
in	<i>handle</i>	Call-back function to handle client requests
in	<i>host</i>	Pointer to a string containing the host address of the server
in	<i>port</i>	Port number of the server
in	<i>ecdsa_priv_key</i>	Buffer containing the ECDSA private key
in	<i>ecdsa_pub_key_x</i>	Buffer containing the x component of the ECDSA public key
in	<i>ecdsa_pub_key_y</i>	Buffer containing the y component of the ECDSA public key

## Returns

Operation status

## Return values

0	Success
<0	Error

**4.8.3.3** `void coap_server_destroy ( coap_server_t * server )`

Deinitialise a server structure.

## Parameters

in, out	<i>server</i>	Pointer to a server structure
---------	---------------	-------------------------------

**4.8.3.4** `unsigned coap_server_get_next_msg_id ( coap_server_t * server )`

Get a new message ID value.

## Parameters

in, out	<i>server</i>	Pointer to a server structure
---------	---------------	-------------------------------

## Returns

message ID value

**4.8.3.5** `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**

<i>in, out</i>	<i>server</i>	Pointer to a server structure
----------------	---------------	-------------------------------

**Returns**

Operation status

**Return values**

<i>0</i>	Success
<i>&lt;0</i>	Error

# Index

active  
    coap\_server\_trans, [15](#)

app\_len  
    coap\_client\_t, [5](#)  
    coap\_server\_trans, [15](#)

app\_start  
    coap\_client\_t, [5](#)  
    coap\_server\_trans, [15](#)

COAP\_CLIENT\_ACK\_TIMEOUT\_SEC  
    coap\_client.c, [42](#)

COAP\_CLIENT\_DTLS\_ALERT  
    coap\_client.h, [19](#)

COAP\_CLIENT\_DTLS\_CONNECTED  
    coap\_client.h, [19](#)

COAP\_CLIENT\_DTLS\_RETRANS\_TIMEOUT  
    coap\_client.c, [42](#)

COAP\_CLIENT\_DTLS\_TOTAL\_TIMEOUT  
    coap\_client.c, [42](#)

COAP\_CLIENT\_DTLS\_UNCONNECTED  
    coap\_client.h, [19](#)

COAP\_CLIENT\_HOST\_BUF\_LEN  
    coap\_client.h, [18](#)

COAP\_CLIENT\_MAX\_RETRANSMIT  
    coap\_client.c, [42](#)

COAP\_CLIENT\_PORT\_BUF\_LEN  
    coap\_client.h, [19](#)

COAP\_CLIENT\_RESP\_TIMEOUT\_SEC  
    coap\_client.c, [42](#)

COAP\_LOG\_DEBUG  
    coap\_log.h, [21](#)

COAP\_LOG\_DEF\_LEVEL  
    coap\_log.h, [21](#)

COAP\_LOG\_ERROR  
    coap\_log.h, [21](#)

COAP\_LOG\_INFO  
    coap\_log.h, [21](#)

COAP\_LOG\_NOTICE  
    coap\_log.h, [21](#)

COAP\_LOG\_WARN  
    coap\_log.h, [21](#)

COAP\_MSG\_ACCEPT  
    coap\_msg.h, [30](#)

COAP\_MSG\_ACK  
    coap\_msg.h, [31](#)

COAP\_MSG\_BAD\_GATEWAY  
    coap\_msg.h, [31](#)

COAP\_MSG\_BAD\_OPTION  
    coap\_msg.h, [30](#)

COAP\_MSG\_BAD\_REQ  
    coap\_msg.h, [30](#)

COAP\_MSG\_CHANGED  
    coap\_msg.h, [31](#)

COAP\_MSG\_CLIENT\_ERR  
    coap\_msg.h, [29](#)

COAP\_MSG\_CON  
    coap\_msg.h, [31](#)

COAP\_MSG\_CONTENT  
    coap\_msg.h, [31](#)

COAP\_MSG\_CONTENT\_FORMAT  
    coap\_msg.h, [30](#)

COAP\_MSG\_CREATED  
    coap\_msg.h, [31](#)

COAP\_MSG\_DELETE  
    coap\_msg.h, [30](#)

COAP\_MSG\_DELETED  
    coap\_msg.h, [31](#)

COAP\_MSG\_ETAG  
    coap\_msg.h, [30](#)

COAP\_MSG\_FORBIDDEN  
    coap\_msg.h, [30](#)

COAP\_MSG\_GATEWAY\_TIMEOUT  
    coap\_msg.h, [31](#)

COAP\_MSG\_GET  
    coap\_msg.h, [30](#)

COAP\_MSG\_IF\_MATCH  
    coap\_msg.h, [30](#)

COAP\_MSG\_IF\_NONE\_MATCH  
    coap\_msg.h, [30](#)

COAP\_MSG\_INT\_SERVER\_ERR  
    coap\_msg.h, [31](#)

COAP\_MSG\_LOCATION\_PATH  
    coap\_msg.h, [30](#)

COAP\_MSG\_LOCATION\_QUERY  
    coap\_msg.h, [30](#)

COAP\_MSG\_MAX\_AGE  
    coap\_msg.h, [30](#)

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\_METHOD\_NOT\_ALLOWED  
    coap\_msg.h, [30](#)

COAP\_MSG\_NON  
     coap\_msg.h, 31  
 COAP\_MSG\_NOT\_ACCEPTABLE  
     coap\_msg.h, 30  
 COAP\_MSG\_NOT\_FOUND  
     coap\_msg.h, 30  
 COAP\_MSG\_NOT\_IMPL  
     coap\_msg.h, 31  
 COAP\_MSG\_OP\_URI\_PATH\_MAX\_LEN  
     coap\_msg.h, 29  
 COAP\_MSG\_OP\_URI\_PATH\_NUM  
     coap\_msg.h, 29  
 COAP\_MSG\_POST  
     coap\_msg.h, 30  
 COAP\_MSG\_PRECOND\_FAILED  
     coap\_msg.h, 30  
 COAP\_MSG\_PROXY\_NOT\_SUP  
     coap\_msg.h, 31  
 COAP\_MSG\_PROXY\_SCHEME  
     coap\_msg.h, 30  
 COAP\_MSG\_PROXY\_URI  
     coap\_msg.h, 30  
 COAP\_MSG\_PUT  
     coap\_msg.h, 30  
 COAP\_MSG\_REQ  
     coap\_msg.h, 29  
 COAP\_MSG\_REQ\_ENT\_TOO\_LARGE  
     coap\_msg.h, 30  
 COAP\_MSG\_RST  
     coap\_msg.h, 31  
 COAP\_MSG\_SERV\_UNAVAIL  
     coap\_msg.h, 31  
 COAP\_MSG\_SERVER\_ERR  
     coap\_msg.h, 29  
 COAP\_MSG\_SIZE1  
     coap\_msg.h, 30  
 COAP\_MSG\_SUCCESS  
     coap\_msg.h, 29  
 COAP\_MSG\_UNAUTHORIZED  
     coap\_msg.h, 30  
 COAP\_MSG\_UNSUP\_CONT\_FMT  
     coap\_msg.h, 30  
 COAP\_MSG\_URI\_HOST  
     coap\_msg.h, 30  
 COAP\_MSG\_URI\_PATH  
     coap\_msg.h, 30  
 COAP\_MSG\_URI\_PORT  
     coap\_msg.h, 30  
 COAP\_MSG\_URI\_QUERY  
     coap\_msg.h, 30  
 COAP\_MSG\_VALID  
     coap\_msg.h, 31  
 COAP\_MSG\_VER  
     coap\_msg.h, 29  
 COAP\_SERVER\_ACK\_TIMEOUT\_SEC  
     coap\_server.c, 54  
 COAP\_SERVER\_ADDR\_BUF\_LEN  
     coap\_server.h, 38  
 COAP\_SERVER\_DIAG\_PAYLOAD\_LEN  
     coap\_server.h, 38  
 COAP\_SERVER\_DTLS\_ALERT  
     coap\_server.h, 39  
 COAP\_SERVER\_DTLS\_CONNECTED  
     coap\_server.h, 39  
 COAP\_SERVER\_DTLS\_MTU  
     coap\_server.c, 54  
 COAP\_SERVER\_DTLS\_NUM\_DH\_BITS  
     coap\_server.c, 54  
 COAP\_SERVER\_DTLS\_PRIORITIES  
     coap\_server.c, 54  
 COAP\_SERVER\_DTLS\_RETRANS\_TIMEOUT  
     coap\_server.c, 54  
 COAP\_SERVER\_DTLS\_TOTAL\_TIMEOUT  
     coap\_server.c, 54  
 COAP\_SERVER\_DTLS\_UNCONNECTED  
     coap\_server.h, 39  
 COAP\_SERVER\_MAX\_RETRANSMIT  
     coap\_server.c, 54  
 COAP\_SERVER\_NUM\_TRANS  
     coap\_server.h, 38  
 COAP\_SERVER\_PIGGYBACKED  
     coap\_server.h, 39  
 COAP\_SERVER\_SEPARATE  
     coap\_server.h, 39  
 client\_addr  
     coap\_server\_trans, 15  
 client\_sin  
     coap\_server\_trans, 15  
 client\_sin\_len  
     coap\_server\_trans, 15  
 coap\_client.c  
     COAP\_CLIENT\_ACK\_TIMEOUT\_SEC, 42  
     COAP\_CLIENT\_DTLS\_RETRANS\_TIMEOUT, 42  
     COAP\_CLIENT\_DTLS\_TOTAL\_TIMEOUT, 42  
     COAP\_CLIENT\_MAX\_RETRANSMIT, 42  
     COAP\_CLIENT\_RESP\_TIMEOUT\_SEC, 42  
     coap\_client\_create, 42  
     coap\_client\_destroy, 43  
     coap\_client\_exchange, 43  
 coap\_client.h  
     COAP\_CLIENT\_DTLS\_ALERT, 19  
     COAP\_CLIENT\_DTLS\_CONNECTED, 19  
     COAP\_CLIENT\_DTLS\_UNCONNECTED, 19  
     COAP\_CLIENT\_HOST\_BUF\_LEN, 18  
     COAP\_CLIENT\_PORT\_BUF\_LEN, 19  
     coap\_client\_create, 19  
     coap\_client\_destroy, 19  
     coap\_client\_dtls\_get\_state, 18  
     coap\_client\_dtls\_state\_t, 19  
     coap\_client\_exchange, 20  
 coap\_client\_create  
     coap\_client.c, 42  
     coap\_client.h, 19  
 coap\_client\_destroy  
     coap\_client.c, 43  
     coap\_client.h, 19

- coap\_client\_dtls\_get\_state
  - coap\_client.h, 18
- coap\_client\_dtls\_state\_t
  - coap\_client.h, 19
- coap\_client\_exchange
  - coap\_client.c, 43
  - coap\_client.h, 20
- coap\_client\_t, 5
  - app\_len, 5
  - app\_start, 5
  - ctx, 5
  - ecdsa\_key, 6
  - num\_retrans, 6
  - sd, 6
  - server\_host, 6
  - server\_port, 6
  - server\_sin, 6
  - server\_sin\_len, 6
  - sess, 6
  - state, 6
  - timeout, 6
  - timer\_fd, 6
- coap\_log.c
  - coap\_log\_debug, 44
  - coap\_log\_error, 44
  - coap\_log\_get\_level, 44
  - coap\_log\_info, 44
  - coap\_log\_notice, 45
  - coap\_log\_set\_level, 45
  - coap\_log\_warn, 45
- coap\_log.h
  - COAP\_LOG\_DEBUG, 21
  - COAP\_LOG\_DEF\_LEVEL, 21
  - COAP\_LOG\_ERROR, 21
  - COAP\_LOG\_INFO, 21
  - COAP\_LOG\_NOTICE, 21
  - COAP\_LOG\_WARN, 21
  - coap\_log\_debug, 21
  - coap\_log\_error, 22
  - coap\_log\_get\_level, 22
  - coap\_log\_info, 22
  - coap\_log\_level\_t, 21
  - coap\_log\_notice, 22
  - coap\_log\_set\_level, 22
  - coap\_log\_warn, 22
- coap\_log\_debug
  - coap\_log.c, 44
  - coap\_log.h, 21
- coap\_log\_error
  - coap\_log.c, 44
  - coap\_log.h, 22
- coap\_log\_get\_level
  - coap\_log.c, 44
  - coap\_log.h, 22
- coap\_log\_info
  - coap\_log.c, 44
  - coap\_log.h, 22
- coap\_log\_level\_t
  - coap\_log.h, 21
- coap\_log\_notice
  - coap\_log.c, 45
  - coap\_log.h, 22
- coap\_log\_set\_level
  - coap\_log.c, 45
  - coap\_log.h, 22
- coap\_log\_warn
  - coap\_log.c, 45
  - coap\_log.h, 22
- coap\_msg.c
  - coap\_msg\_add\_op, 47
  - coap\_msg\_check\_critical\_ops, 47
  - coap\_msg\_check\_unsafe\_ops, 48
  - coap\_msg\_copy, 48
  - coap\_msg\_create, 48
  - coap\_msg\_destroy, 49
  - coap\_msg\_format, 49
  - coap\_msg\_gen\_rand\_str, 49
  - coap\_msg\_op\_list\_get\_first, 47
  - coap\_msg\_op\_list\_get\_last, 47
  - coap\_msg\_op\_list\_is\_empty, 47
  - coap\_msg\_op\_num\_is\_recognized, 49
  - coap\_msg\_parse, 50
  - coap\_msg\_parse\_type\_msg\_id, 50
  - coap\_msg\_reset, 50
  - coap\_msg\_set\_code, 51
  - coap\_msg\_set\_msg\_id, 51
  - coap\_msg\_set\_payload, 51
  - coap\_msg\_set\_token, 52
  - coap\_msg\_set\_type, 52
- coap\_msg.h
  - COAP\_MSG\_ACCEPT, 30
  - COAP\_MSG\_ACK, 31
  - COAP\_MSG\_BAD\_GATEWAY, 31
  - COAP\_MSG\_BAD\_OPTION, 30
  - COAP\_MSG\_BAD\_REQ, 30
  - COAP\_MSG\_CHANGED, 31
  - COAP\_MSG\_CLIENT\_ERR, 29
  - COAP\_MSG\_CON, 31
  - COAP\_MSG\_CONTENT, 31
  - COAP\_MSG\_CONTENT\_FORMAT, 30
  - COAP\_MSG\_CREATED, 31
  - COAP\_MSG\_DELETE, 30
  - COAP\_MSG\_DELETED, 31
  - COAP\_MSG\_ETAG, 30
  - COAP\_MSG\_FORBIDDEN, 30
  - COAP\_MSG\_GATEWAY\_TIMEOUT, 31
  - COAP\_MSG\_GET, 30
  - COAP\_MSG\_IF\_MATCH, 30
  - COAP\_MSG\_IF\_NONE\_MATCH, 30
  - COAP\_MSG\_INT\_SERVER\_ERR, 31
  - COAP\_MSG\_LOCATION\_PATH, 30
  - COAP\_MSG\_LOCATION\_QUERY, 30
  - COAP\_MSG\_MAX\_AGE, 30
  - 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\_METHOD\_NOT\_ALLOWED, 30  
 COAP\_MSG\_NON, 31  
 COAP\_MSG\_NOT\_ACCEPTABLE, 30  
 COAP\_MSG\_NOT\_FOUND, 30  
 COAP\_MSG\_NOT\_IMPL, 31  
 COAP\_MSG\_OP\_URI\_PATH\_MAX\_LEN, 29  
 COAP\_MSG\_OP\_URI\_PATH\_NUM, 29  
 COAP\_MSG\_POST, 30  
 COAP\_MSG\_PRECOND\_FAILED, 30  
 COAP\_MSG\_PROXY\_NOT\_SUP, 31  
 COAP\_MSG\_PROXY\_SCHEME, 30  
 COAP\_MSG\_PROXY\_URI, 30  
 COAP\_MSG\_PUT, 30  
 COAP\_MSG\_REQ, 29  
 COAP\_MSG\_REQ\_ENT\_TOO\_LARGE, 30  
 COAP\_MSG\_RST, 31  
 COAP\_MSG\_SERV\_UNAVAIL, 31  
 COAP\_MSG\_SERVER\_ERR, 29  
 COAP\_MSG\_SIZE1, 30  
 COAP\_MSG\_SUCCESS, 29  
 COAP\_MSG\_UNAUTHORIZED, 30  
 COAP\_MSG\_UNSUP\_CONT\_FMT, 30  
 COAP\_MSG\_URI\_HOST, 30  
 COAP\_MSG\_URI\_PATH, 30  
 COAP\_MSG\_URI\_PORT, 30  
 COAP\_MSG\_URI\_QUERY, 30  
 COAP\_MSG\_VALID, 31  
 COAP\_MSG\_VER, 29  
 coap\_msg\_add\_op, 31  
 coap\_msg\_check\_critical\_ops, 32  
 coap\_msg\_check\_unsafe\_ops, 32  
 coap\_msg\_class\_t, 29  
 coap\_msg\_client\_err\_t, 29  
 coap\_msg\_copy, 32  
 coap\_msg\_create, 33  
 coap\_msg\_destroy, 33  
 coap\_msg\_format, 33  
 coap\_msg\_gen\_rand\_str, 33  
 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, 27  
 coap\_msg\_is\_empty, 28  
 coap\_msg\_method\_t, 30  
 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\_recognized, 34  
 coap\_msg\_op\_num\_is\_unsafe, 28  
 coap\_msg\_op\_num\_no\_cache\_key, 29  
 coap\_msg\_op\_num\_t, 30  
 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, 34  
 coap\_msg\_parse\_type\_msg\_id, 34  
 coap\_msg\_reset, 35  
 coap\_msg\_server\_err\_t, 30  
 coap\_msg\_set\_code, 35  
 coap\_msg\_set\_msg\_id, 35  
 coap\_msg\_set\_payload, 35  
 coap\_msg\_set\_token, 36  
 coap\_msg\_set\_type, 36  
 coap\_msg\_success\_t, 31  
 coap\_msg\_type\_t, 31  
 coap\_msg\_add\_op  
     coap\_msg.c, 47  
     coap\_msg.h, 31  
 coap\_msg\_check\_critical\_ops  
     coap\_msg.c, 47  
     coap\_msg.h, 32  
 coap\_msg\_check\_unsafe\_ops  
     coap\_msg.c, 48  
     coap\_msg.h, 32  
 coap\_msg\_class\_t  
     coap\_msg.h, 29  
 coap\_msg\_client\_err\_t  
     coap\_msg.h, 29  
 coap\_msg\_copy  
     coap\_msg.c, 48  
     coap\_msg.h, 32  
 coap\_msg\_create  
     coap\_msg.c, 48  
     coap\_msg.h, 33  
 coap\_msg\_destroy  
     coap\_msg.c, 49  
     coap\_msg.h, 33  
 coap\_msg\_format  
     coap\_msg.c, 49  
     coap\_msg.h, 33  
 coap\_msg\_gen\_rand\_str  
     coap\_msg.c, 49  
     coap\_msg.h, 33  
 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, [27](#)
- coap\_msg\_is\_empty
  - coap\_msg.h, [28](#)
- coap\_msg\_method\_t
  - coap\_msg.h, [30](#)
- coap\_msg\_op, [7](#)
  - len, [7](#)
  - next, [7](#)
  - num, [7](#)
  - val, [7](#)
- 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, [47](#)
- coap\_msg\_op\_list\_get\_last
  - coap\_msg.c, [47](#)
- coap\_msg\_op\_list\_is\_empty
  - coap\_msg.c, [47](#)
- coap\_msg\_op\_list\_t, [8](#)
  - first, [8](#)
  - last, [8](#)
- coap\_msg\_op\_num\_is\_critical
  - coap\_msg.h, [28](#)
- coap\_msg\_op\_num\_is\_recognized
  - coap\_msg.c, [49](#)
  - coap\_msg.h, [34](#)
- coap\_msg\_op\_num\_is\_unsafe
  - coap\_msg.h, [28](#)
- coap\_msg\_op\_num\_no\_cache\_key
  - coap\_msg.h, [29](#)
- coap\_msg\_op\_num\_t
  - coap\_msg.h, [30](#)
- 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, [50](#)
  - coap\_msg.h, [34](#)
- coap\_msg\_parse\_type\_msg\_id
  - coap\_msg.c, [50](#)
  - coap\_msg.h, [34](#)
- coap\_msg\_reset
  - coap\_msg.c, [50](#)
  - coap\_msg.h, [35](#)
- coap\_msg\_server\_err\_t
  - coap\_msg.h, [30](#)
- coap\_msg\_set\_code
  - coap\_msg.c, [51](#)
  - coap\_msg.h, [35](#)
- coap\_msg\_set\_msg\_id
  - coap\_msg.c, [51](#)
  - coap\_msg.h, [35](#)
- coap\_msg\_set\_payload
  - coap\_msg.c, [51](#)
  - coap\_msg.h, [35](#)
- coap\_msg\_set\_token
  - coap\_msg.c, [52](#)
  - coap\_msg.h, [36](#)
- coap\_msg\_set\_type
  - coap\_msg.c, [52](#)
  - coap\_msg.h, [36](#)
- coap\_msg\_success\_t
  - coap\_msg.h, [31](#)
- coap\_msg\_t, [8](#)
  - code\_class, [9](#)
  - code\_detail, [9](#)
  - msg\_id, [9](#)
  - op\_list, [9](#)
  - payload, [10](#)
  - payload\_len, [10](#)
  - token, [10](#)
  - token\_len, [10](#)
  - type, [10](#)
  - ver, [10](#)
- coap\_msg\_type\_t
  - coap\_msg.h, [31](#)
- coap\_server, [10](#)
  - ecdsa\_key, [11](#)
  - handle, [11](#)
  - msg\_id, [11](#)
  - sd, [12](#)
  - sep\_list, [12](#)
  - trans, [12](#)
- coap\_server.c
  - COAP\_SERVER\_ACK\_TIMEOUT\_SEC, [54](#)
  - COAP\_SERVER\_DTLS\_MTU, [54](#)
  - COAP\_SERVER\_DTLS\_NUM\_DH\_BITS, [54](#)
  - COAP\_SERVER\_DTLS\_PRIORITIES, [54](#)
  - COAP\_SERVER\_DTLS\_RETRANS\_TIMEOUT, [54](#)
  - COAP\_SERVER\_DTLS\_TOTAL\_TIMEOUT, [54](#)
  - COAP\_SERVER\_MAX\_RETRANSMIT, [54](#)
  - coap\_server\_add\_sep\_resp\_uri\_path, [54](#)
  - coap\_server\_create, [55](#)
  - coap\_server\_destroy, [55](#)
  - coap\_server\_get\_next\_msg\_id, [55](#)
  - coap\_server\_run, [55](#)
- coap\_server.h

- COAP\_SERVER\_ADDR\_BUF\_LEN, 38
- COAP\_SERVER\_DIAG\_PAYLOAD\_LEN, 38
- COAP\_SERVER\_DTLS\_ALERT, 39
- COAP\_SERVER\_DTLS\_CONNECTED, 39
- COAP\_SERVER\_DTLS\_UNCONNECTED, 39
- COAP\_SERVER\_NUM\_TRANS, 38
- COAP\_SERVER\_PIGGYBACKED, 39
- COAP\_SERVER\_SEPARATE, 39
- coap\_server\_add\_sep\_resp\_uri\_path, 39
- coap\_server\_create, 39
- coap\_server\_destroy, 40
- coap\_server\_dtls\_state\_t, 39
- coap\_server\_get\_next\_msg\_id, 40
- coap\_server\_resp\_t, 39
- coap\_server\_run, 40
- coap\_server\_add\_sep\_resp\_uri\_path
  - coap\_server.c, 54
  - coap\_server.h, 39
- coap\_server\_create
  - coap\_server.c, 55
  - coap\_server.h, 39
- coap\_server\_destroy
  - coap\_server.c, 55
  - coap\_server.h, 40
- coap\_server\_dtls\_state\_t
  - coap\_server.h, 39
- coap\_server\_get\_next\_msg\_id
  - coap\_server.c, 55
  - coap\_server.h, 40
- coap\_server\_path, 12
  - next, 12
  - str, 12
- coap\_server\_path\_list\_t, 13
  - first, 13
  - last, 13
- coap\_server\_resp\_t
  - coap\_server.h, 39
- coap\_server\_run
  - coap\_server.c, 55
  - coap\_server.h, 40
- coap\_server\_trans, 14
  - active, 15
  - app\_len, 15
  - app\_start, 15
  - client\_addr, 15
  - client\_sin, 15
  - client\_sin\_len, 15
  - ctx, 15
  - last\_use, 15
  - num\_retrans, 15
  - req, 15
  - resp, 15
  - server, 16
  - sess, 16
  - state, 16
  - timeout, 16
  - timer\_fd, 16
- code\_class
  - coap\_msg\_t, 9
- code\_detail
  - coap\_msg\_t, 9
- ctx
  - coap\_client\_t, 5
  - coap\_server\_trans, 15
- ecdsa\_key
  - coap\_client\_t, 6
  - coap\_server, 11
- first
  - coap\_msg\_op\_list\_t, 8
  - coap\_server\_path\_list\_t, 13
- handle
  - coap\_server, 11
- last
  - coap\_msg\_op\_list\_t, 8
  - coap\_server\_path\_list\_t, 13
- last\_use
  - coap\_server\_trans, 15
- len
  - coap\_msg\_op, 7
- lib/include/coap\_client.h, 17
- lib/include/coap\_log.h, 20
- lib/include/coap\_msg.h, 24
- lib/include/coap\_server.h, 36
- lib/src/coap\_client.c, 41
- lib/src/coap\_log.c, 43
- lib/src/coap\_msg.c, 45
- lib/src/coap\_server.c, 52
- msg\_id
  - coap\_msg\_t, 9
  - coap\_server, 11
- next
  - coap\_msg\_op, 7
  - coap\_server\_path, 12
- num
  - coap\_msg\_op, 7
- num\_retrans
  - coap\_client\_t, 6
  - coap\_server\_trans, 15
- op\_list
  - coap\_msg\_t, 9
- payload
  - coap\_msg\_t, 10
- payload\_len
  - coap\_msg\_t, 10
- req
  - coap\_server\_trans, 15
- resp
  - coap\_server\_trans, 15
- sd



- coap\_client\_t, [6](#)
  - coap\_server, [12](#)
- sep\_list
  - coap\_server, [12](#)
- server
  - coap\_server\_trans, [16](#)
- server\_host
  - coap\_client\_t, [6](#)
- server\_port
  - coap\_client\_t, [6](#)
- server\_sin
  - coap\_client\_t, [6](#)
- server\_sin\_len
  - coap\_client\_t, [6](#)
- sess
  - coap\_client\_t, [6](#)
  - coap\_server\_trans, [16](#)
- state
  - coap\_client\_t, [6](#)
  - coap\_server\_trans, [16](#)
- str
  - coap\_server\_path, [12](#)
- timeout
  - coap\_client\_t, [6](#)
  - coap\_server\_trans, [16](#)
- timer\_fd
  - coap\_client\_t, [6](#)
  - coap\_server\_trans, [16](#)
- token
  - coap\_msg\_t, [10](#)
- token\_len
  - coap\_msg\_t, [10](#)
- trans
  - coap\_server, [12](#)
- type
  - coap\_msg\_t, [10](#)
- val
  - coap\_msg\_op, [7](#)
- ver
  - coap\_msg\_t, [10](#)