

sl_mqtt_server

[Main Page](#)

[Modules](#)

[Classes](#)

[Files](#)

Modules

Here is a list of all modules:

- [SL MQTT Server APIs](#)
 - [SL MQTT Server Events](#)

Generated on Thu Jan 15 2015 18:28:55 for sl_mqtt_server by
[doxygen](#) 1.8.0

sl_mqtt_server

[Main Page](#)

[Modules](#)

[Classes](#)

[Files](#)

[Modules](#) | [Classes](#) | [Functions](#)

SL MQTT Server APIs

Modules

SL MQTT Server Events

Classes

```
struct SIMqttServerCbs_t  
struct SIMqttServerParams_t  
struct SIMqttServerCfg_t
```

Functions

```
_i32 sl_ExtLib_MqttServerInit (_const SIMqttServerCfg_t *cfg,  
                               _const SIMqttServerCbs_t *cbs)  
_i32 sl_ExtLib_MqttTopicEnroll (_const char *topic)  
_i32 sl_ExtLib_MqttTopicDisenroll (_const char *topic)  
_i32 sl_ExtLib_MqttServerSend (_const char *topic, _const void  
                               *data, _i32 len, _u8 qos, bool retain, _u32 flags)
```

Function Documentation

```
_i32 sl_ExtLib_MqttServerInit( _const SIMqttServerCfg_t * cfg,  
                               _const SIMqttServerCbs_t * cbs  
                           )
```

Initialize the SL MQTT Server Implementation. A caller must initialize the MQTT Server implementation prior to using its services.

Parameters:

[in] **cfg** refers to configuration parameters
[in] **cbs** callbacks into server application

Returns:

Success (0) or Failure (-1)

```
_i32 sl_ExtLib_MqttServerSend( _const char * topic,  
                               _const void * data,  
                               _i32           len,  
                               _u8            qos,  
                               bool           retain,  
                               _u32           flags  
                           )
```

PUBLISH a named message to a client. In addition to the PUBLISH specific parameters, the caller can indicate whether the routine should block until the time the message has been acknowledged by the client. This is applicable only for non-QoS0 messages.

In case, the app has chosen not to await for the ACK from the client, the SL MQTT Server implementation will notify the app about the received ACK through the callback routine.

Note:

Only QoS0 and QoS1 messages are supported.

Parameters:

- [in] **topic** topic of the data to be published. It is NULL terminated.
- [in] **data** binary data to be published
- [in] **len** length of the data
- [in] **qos** QoS for the publish message
- [in] **retain** Retain bit in the PUBLISH message sent out
- [in] **flags** Reserved for future use.

Returns:

Success(0) or Failure(-1).

_i32 sl_ExtLib_MqttTopicDisenroll (_const char * topic)

DisEnroll a topic. App can cancel the previous enrollment of a topic and the SL layer will now stop forwarding data published for that topic to the server by the connected clients. This is analogous oe similar to an unsubscription by an MQTT client.

Parameters:

- [in] **topic** disenrollment topic. It is NULL terminated.

Returns:

Success(0) or Failure(-1).

_i32 sl_ExtLib_MqttTopicEnroll (_const char * topic)

Enroll a topic to receive data App can enroll a topic of interest and the SL layer will forward to the app any data subsequently published for the enrolled topic to the server by any of the connected MQTT clients.

This is analogous or similar to a subscription to a topic by an MQTT client.

Parameters:

[in] **topic** enrollment topic for data to be received. It is NULL terminated.

Returns:

Success(0) or Failure (-1).

sl_mqtt_server

[Main Page](#)

[Modules](#)

[Classes](#)

[Files](#)

Defines

SL MQTT Server Events

[SL MQTT Server APIs](#)

Defines

```
#define SL_MQTT_SRVR_EVT_PUBACK 0x11
#define SL_MQTT_SRVR_EVT_NOCONN 0x12
```

Define Documentation

#define SL_MQTT_SRVR_EVT_NOCONN 0x12

Server has lost network connection with a client

#define SL_MQTT_SRVR_EVT_PUBACK 0x11

PUBACK has been received from a client

Generated on Thu Jan 15 2015 18:28:55 for sl_mqtt_server by
[doxygen](#) 1.8.0

sl_mqtt_server

Main Page	Modules	Classes	Files	
Class List	Class Index	Class Members		Public Attributes

SI~~M~~qttServerCbs_t Struct Reference

[SL MQTT Server APIs](#)

```
#include <sl_mqtt_server.h>
```

List of all members.

Public Attributes

```
_u16(* sl_ExtLib_MqttConn )(_const char *clientId_str, _i32
                           clientId_len, _const char *username_str, _i32 username_len,
                           _const char *password_str, _i32 password_len, void **usr)
    void(* sl_ExtLib_MqttRecv )(_const char *topstr, _i32 toplen, _const
                               void *payload, _i32 pay_len, bool dup, _u8 qos, bool retain)
    void(* sl_ExtLib_MqttDisconnect )(void *usr, bool due2err)
    void(* sl_ExtLib_MqttEvent )(void *usr, _i32 evt, _const void *buf,
                                  _u32 len)
```

Detailed Description

Callbacks Routines The routines are invoked by SL MQTT Server Implementation onto Server application

Note:

The user applications implement the callbacks that are registered with the libraries. While using the MQTT library, invoking the core library APIs from a callback should be avoided and can lead to lockup scenarios. It is recommended to signal another task from the callback routines invoked from the library and invoke the core library API calls from that task.

Member Data Documentation

`_u16(* SIMqttServerCbs_t::sl_ExtLib_MqttConn)(_const char *clientID, _const char *username, _const char *password, _const char *willTopic, _const char *willMessage, _const char *willQoS, _const char *willRetain, void *usr, bool due2err)`

Connect Request: Callback routine to indicate to server application that a CONNECT request has been received by the server.

```
\param[in] clientID_str ClientID field in the CONNECT message received.  
\param[in] clientID_len length of ClientID  
\param[in] username_str Username field in the CONNECT message received.  
\param[in] username_len length of username  
\param[in] password_str Password field in the CONNECT message received.  
\param[in] password_len length of password  
\param[out] usr placeholder to provision app's handle to this connection.  
\return 0x0000 or 0x0100 for success; Otherwise 0x02, 0x04 or 0x05  
refer to spec for connack codes
```

`void(* SIMqttServerCbs_t::sl_ExtLib_Disconn)(void *usr, bool due2err)`

DISCONNECT: Callback routine to indicate to the Server Application that a client has disconnected.

Parameters:

- [in] **usr** app's handle to this connection.
- [in] **due2err** set to 1, if connection has been closed, without server receiving a DISCONNECT message.

Returns:

none.

void(* SIMqttServerCbs_t::sl_ExtLib_MqttEvent)(void *usr, _i32 evt

Indication of event either from the server library or implementation generated. TBD - Reserved for future use.

Parameters:

- [in] **usr** app's handle to this connection.
- [in] **evt** identifier to the reported event. Refer to [SL MQTT Server Events](#) - TBD
- [in] **buf** points to buffer
- [in] **len** length of buffer

Returns:

none.

void(* SIMqttServerCbs_t::sl_ExtLib_MqttRecv)(_const char *topst

Callback routine to receive a PUBLISH from a client. The server app must provide this routine for the instances where it receives PUBLISH messages from clients. The callback is invoked in the context of the internal SL MQTT server Receive Task.

Parameters:

- [in] **topstr** name of topic on which PUBLISH is received by the server. Not NULL terminated.
- [in] **toplen** length of the topic name
- [in] **payload** refers to payload published by the server.
- [in] **pay_len** length of the payload.
- [in] **dup** assert to indicate that it is a duplicate message sent by the client
- [in] **qoS** quality of service of the received published message.
- [in] **retain** asserted to indicate that a retained message has been received

Returns:

none.

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

- D:/Project/SimpleLink/mqtt/doxygen/server/**sl_mqtt_server.h**
-

Generated on Thu Jan 15 2015 18:28:55 for sl_mqtt_server by
doxygen 1.8.0

sl_mqtt_server

Main Page	Modules	Classes	Files
Class List	Class Index	Class Members	
			Public Attributes

SI~~M~~qttServerParams_t Struct Reference

[SL MQTT Server APIs](#)

#include <[sl_mqtt_server.h](#)>

List of all members.

Public Attributes

```
_u32 netconn_flags  
_const char * server_addr  
_u16 port_number  
_u8 method  
_u32 cipher  
_u32 n_files  
char *_const * secure_files
```

Detailed Description

Secure Socket Parameters to open a secure connection

Member Data Documentation

_u32 SIMqttServerParams_t::cipher

Cipher to tcp secured socket

_u8 SIMqttServerParams_t::method

Method to tcp secured socket

_u32 SIMqttServerParams_t::n_files

Number of files for secure transfer

_u32 SIMqttServerParams_t::netconn_flags

Enumerate connection type

_u16 SIMqttServerParams_t::port_number

Port number of MQTT server

char* _const* SIMqttServerParams_t::secure_files

_const char* SIMqttServerParams_t::server_addr

Server Address: URL or IP

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

- D:/Project/SimpleLink/mqtt/doxygen/server/**sl_mqtt_server.h**
-

Generated on Thu Jan 15 2015 18:28:55 for **sl_mqtt_server** by
doxygen 1.8.0

sl_mqtt_server

Main Page	Modules	Classes	Files	
Class List	Class Index	Class Members		Public Attributes

SI~~M~~qttServerCfg_t Struct Reference

[SL MQTT Server APIs](#)

```
#include <sl_mqtt_server.h>
```

List of all members.

Public Attributes

```
SIMqttServerParams_t server_info
    _u16 loopback_port
    _u32 rx_tsk_priority
    _u32 resp_time
    bool aux_debug_en
    _i32(* dbg_print )(_const char *pcFormat,...)
```

Detailed Description

MQTT Server Lib structure which holds Initialization Data

Member Data Documentation

`bool SIMqttServerCfg_t::aux_debug_en`

Assert to indicate additional debug info

`_i32(* SIMqttServerCfg_t::dbg_print)(_const char *pcFormat,...)`

Print debug information

`_u16 SIMqttServerCfg_t::loopback_port`

Loopback port to manage lib internal functioning - Mandatory

`_u32 SIMqttServerCfg_t::resp_time`

Reasonable response wait time (seconds) for server

`_u32 SIMqttServerCfg_t::rx_tsk_priority`

Priority of the receive task in server

`SIMqttServerParams_t SIMqttServerCfg_t::server_info`

Server information

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

- D:/Project/SimpleLink/mqtt/doxygen/server/**sl_mqtt_server.h**

Generated on Thu Jan 15 2015 18:28:55 for sl_mqtt_server by
[doxygen](#) 1.8.0

sl_mqtt_server

Main Page	Modules	Classes	Files
Class List	Class Index	Class Members	

Class List

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

[SIMqttServerCbs_t](#)

[SIMqttServerCfg_t](#)

[SIMqttServerParams_t](#)

Generated on Thu Jan 15 2015 18:28:55 for sl_mqtt_server by
[doxygen](#) 1.8.0

sl_mqtt_server

Main Page	Modules	Classes	Files
Class List	Class Index	Class Members	

Class Index

S

S

[SIMqttServerCfg_t](#) [SIMqttServerParams_t](#)

[SIMqttServerCbs_t](#)

S

Generated on Thu Jan 15 2015 18:28:55 for sl_mqtt_server by
[doxygen](#) 1.8.0

sl_mqtt_server

Main Page	Modules	Classes	Files
Class List	Class Index	Class Members	
All	Variables		

Here is a list of all class members with links to the classes they belong to:

- aux_debug_en : [SIMqttServerCfg_t](#)
- cipher : [SIMqttServerParams_t](#)
- dbg_print : [SIMqttServerCfg_t](#)
- loopback_port : [SIMqttServerCfg_t](#)
- method : [SIMqttServerParams_t](#)
- n_files : [SIMqttServerParams_t](#)
- netconn_flags : [SIMqttServerParams_t](#)
- port_number : [SIMqttServerParams_t](#)
- resp_time : [SIMqttServerCfg_t](#)
- rx_tsk_priority : [SIMqttServerCfg_t](#)
- secure_files : [SIMqttServerParams_t](#)
- server_addr : [SIMqttServerParams_t](#)
- server_info : [SIMqttServerCfg_t](#)
- sl_ExtLib_MqttConn : [SIMqttServerCbs_t](#)
- sl_ExtLib_MqttDisconnect : [SIMqttServerCbs_t](#)
- sl_ExtLib_MqttEvent : [SIMqttServerCbs_t](#)
- sl_ExtLib_MqttRecv : [SIMqttServerCbs_t](#)

sl_mqtt_server

Main Page	Modules	Classes	Files
Class List	Class Index	Class Members	
All	Variables		

- aux_debug_en : [SIMqttServerCfg_t](#)
- cipher : [SIMqttServerParams_t](#)
- dbg_print : [SIMqttServerCfg_t](#)
- loopback_port : [SIMqttServerCfg_t](#)
- method : [SIMqttServerParams_t](#)
- n_files : [SIMqttServerParams_t](#)
- netconn_flags : [SIMqttServerParams_t](#)
- port_number : [SIMqttServerParams_t](#)
- resp_time : [SIMqttServerCfg_t](#)
- rx_tsk_priority : [SIMqttServerCfg_t](#)
- secure_files : [SIMqttServerParams_t](#)
- server_addr : [SIMqttServerParams_t](#)
- server_info : [SIMqttServerCfg_t](#)
- sl_ExtLib_MqttConn : [SIMqttServerCbs_t](#)
- sl_ExtLib_MqttDisconnect : [SIMqttServerCbs_t](#)
- sl_ExtLib_MqttEvent : [SIMqttServerCbs_t](#)
- sl_ExtLib_MqttRecv : [SIMqttServerCbs_t](#)

sl_mqtt_server

Main Page

Modules

Classes

Files

File List

File Members

File List

Here is a list of all files with brief descriptions:

[D:/Project/SimpleLink/mqtt/doxygen/server/sl_mqtt_server.h](#) [code]

Generated on Thu Jan 15 2015 18:28:55 for sl_mqtt_server by
[doxygen](#) 1.8.0

sl_mqtt_server

[Main Page](#)[Modules](#)[Classes](#)[Files](#)[File List](#)[File Members](#)[Classes](#) | [Defines](#) | [Functions](#)

D:/Project/SimpleLink/mqtt/doxygen/server/sl_m File Reference

```
#include <stdio.h> #include <string.h>
#include <stdbool.h>
#include "simplelink.h"
```

[Go to the source code of this file.](#)

Classes

```
struct SIMqttServerCbs_t  
struct SIMqttServerParams_t  
struct SIMqttServerCfg_t
```

Defines

```
#define SL_MQTT_SRVR_EVT_PUBACK 0x11
#define SL_MQTT_SRVR_EVT_NOCONN 0x12
#define SL_MQTT_SRVR_NETCONN_IP6 0x01
#define SL_MQTT_SRVR_NETCONN_URL 0x02
#define SL_MQTT_SRVR_NETCONN_SEC 0x04
```

Functions

```
_i32 sl_ExtLib_MqttServerInit (_const SIMqttServerCfg_t *cfg,  
                               _const SIMqttServerCbs_t *cbs)  
_i32 sl_ExtLib_MqttTopicEnroll (_const char *topic)  
_i32 sl_ExtLib_MqttTopicDisenroll (_const char *topic)  
_i32 sl_ExtLib_MqttServerSend (_const char *topic, _const void  
                               *data, _i32 len, _u8 qos, bool retain, _u32 flags)
```

Define Documentation

#define SL_MQTT_SRVR_NETCONN_IP6 0x01

Assert for IPv6 connection, otherwise IPv4

#define SL_MQTT_SRVR_NETCONN_SEC 0x04

Connection to server must be secure (TLS)

#define SL_MQTT_SRVR_NETCONN_URL 0x02

Server address is an URL and not IP address

Generated on Thu Jan 15 2015 18:28:55 for sl_mqtt_server by
[doxygen](#) 1.8.0

sl_mqtt_server

Main Page	Modules	Classes	Files
File List	File Members		
All	Functions	Defines	

Here is a list of all file members with links to the files they belong to:

- sl_ExtLib_MqttServerInit() : [sl_mqtt_server.h](#)
- sl_ExtLib_MqttServerSend() : [sl_mqtt_server.h](#)
- sl_ExtLib_MqttTopicDisenroll() : [sl_mqtt_server.h](#)
- sl_ExtLib_MqttTopicEnroll() : [sl_mqtt_server.h](#)
- SL_MQTT_SRVR_EVT_NOCONN : [sl_mqtt_server.h](#)
- SL_MQTT_SRVR_EVT_PUBACK : [sl_mqtt_server.h](#)
- SL_MQTT_SRVR_NETCONN_IP6 : [sl_mqtt_server.h](#)
- SL_MQTT_SRVR_NETCONN_SEC : [sl_mqtt_server.h](#)
- SL_MQTT_SRVR_NETCONN_URL : [sl_mqtt_server.h](#)

Generated on Thu Jan 15 2015 18:28:55 for sl_mqtt_server by
[doxygen](#) 1.8.0

sl_mqtt_server

Main Page	Modules	Classes	Files
File List	File Members		
All	Functions	Defines	

- sl_ExtLib_MqttServerInit() : [sl_mqtt_server.h](#)
- sl_ExtLib_MqttServerSend() : [sl_mqtt_server.h](#)
- sl_ExtLib_MqttTopicDisenroll() : [sl_mqtt_server.h](#)
- sl_ExtLib_MqttTopicEnroll() : [sl_mqtt_server.h](#)

Generated on Thu Jan 15 2015 18:28:55 for sl_mqtt_server by
[doxygen](#) 1.8.0

sl_mqtt_server

Main Page	Modules	Classes	Files
File List	File Members		
All	Functions	Defines	

- SL_MQTT_SRVR_EVT_NOCONN : [sl_mqtt_server.h](#)
- SL_MQTT_SRVR_EVT_PUBACK : [sl_mqtt_server.h](#)
- SL_MQTT_SRVR_NETCONN_IP6 : [sl_mqtt_server.h](#)
- SL_MQTT_SRVR_NETCONN_SEC : [sl_mqtt_server.h](#)
- SL_MQTT_SRVR_NETCONN_URL : [sl_mqtt_server.h](#)

Generated on Thu Jan 15 2015 18:28:55 for sl_mqtt_server by
[doxygen](#) 1.8.0

sl_mqtt_server

Main Page	Modules	Classes	Files
File List	File Members		

D:/Project/SimpleLink/mqtt/doxygen/server/sl_m

Go to the documentation of this file.

```
00001 /*****
* ****
00002 Copyright (c) (2014) Texas Instruments Incorporated
00003 All rights reserved not granted herein.
00004
00005 Limited License.
00006
00007 Texas Instruments Incorporated grants a world-wide, royalty-free, non-exclusive
00008 license under copyrights and patents it now or hereafter owns or controls to make,
00009 have made, use, import, offer to sell and sell ("Utilize") this software subject
00010 to the terms herein. With respect to the foregoing patent license, such license
00011 is granted solely to the extent that any such patent is necessary to Utilize the
00012 software alone. The patent license shall not apply to any combinations which
00013 include this software, other than combinations with devices manufactured by or
00014 for TI (TI Devices). No hardware patent is licensed hereunder.
00015
00016 Redistributions must preserve existing copyright notices and reproduce this license
```

00017 (including the above copyright notice and th
e disclaimer and (if applicable) source
00018 code license limitations below) in the docum
entation and/or other materials provided
00019 with the distribution
00020
00021 Redistribution and use in binary form, witho
ut modification, are permitted provided
00022 that the following conditions are met:
00023 * No reverse engineering, decompilation, or
disassembly of this software is
00024 permitted with respect to any software pro
vided in binary form.
00025 * any redistribution and use are licensed by
TI for use only with TI Devices.
00026 * Nothing shall obligate TI to provide you w
ith source code for the software
00027 licensed and provided to you in object cod
e.
00028
00029 If software source code is provided to you,
modification and redistribution of the
00030 source code are permitted provided that the
following conditions are met;
00031 * any redistribution and use of the source c
ode, including any resulting derivative
00032 works, are licensed by TI for use only wit
h TI Devices.
00033 * any redistribution and use of any object c
ode compiled from the source code and
00034 any resulting derivative works, are licens
ed by TI for use only with TI Devices.
00035
00036 Neither the name of Texas Instruments Incorp
orated nor the names of its suppliers
00037 may be used to endorse or promote products d
erived from this software without

00038 specific prior written permission.

00039

00040 DISCLAIMER.

00041

00042 THIS SOFTWARE IS PROVIDED BY TI AND TI'S LIC
ENSORS "AS IS" AND ANY EXPRESS OR IMPLIED

00043 WARRANTIES, INCLUDING, BUT NOT LIMITED TO, T
HE IMPLIED WARRANTIES OF MERCHANTABILITY

00044 AND FITNESS FOR A PARTICULAR PURPOSE ARE DIS
CLAIMED. IN NO EVENT SHALL TI AND TI'S

00045 LICENSORS BE LIABLE FOR ANY DIRECT, INDIRECT
, INCIDENTAL, SPECIAL, EXEMPLARY, OR

00046 CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LI
MITED TO, PROCUREMENT OF SUBSTITUTE

00047 GOODS OR SERVICES; LOSS OF USE, DATA, OR PRO
FITS; OR BUSINESS INTERRUPTION)

00048 HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY
, WHETHER IN CONTRACT, STRICT LIABILITY,

00049 OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF

00050 THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIB
ILITY OF SUCH DAMAGE.

00051

00052 *****/

00053

00054 #include <stdio.h>

00055 #include <string.h>

00056 #include <stdbool.h>

00057 #include "simplelink.h"

00058

00059 #ifndef __SL_MQTT_SRVR_H__

00060 #define __SL_MQTT_SRVR_H__

00061

00062 #ifdef __cplusplus

00063 extern "C"

00064 {

```
00065 #endif
00066
00126 #define SL_MQTT_SRVR_EVT_PUBACK      0x11
00127 #define SL_MQTT_SRVR_EVT_NOCONN     0x12    /*
End SL MQTT Server events */
00129
00130
00141     typedef struct {
00142
00157             _u16 (*sl_ExtLib_MqttConn)(
00158     _const char *clientId_str, _i32 clientId_len,
00159     _const char *username_str, _i32 username_len,
00160     _const char *password_str, _i32 password_len,
00161     void **usr);
00162
00177             void (*sl_ExtLib_MqttRecv)(_
00178     _const char *topstr,    _i32 toplen,
00179     _const void *payload,   _i32 pay_len,
00180     bool dup,   _u8 qos,
00181     bool retain);
00182             // Note: Double check whether
00183             // retain, dup, qos would be required for a server
00184             // app (with no topic management)
00185
00192             void (*sl_ExtLib_MqttDisconnect(
00193     void *usr, bool due2err));
00194
00204             void (*sl_ExtLib_MqttEvent)(
```

```

void *usr, _i32 evt, _const void *buf, _u32 len);
00205
00206         } SlMqttServerCbs_t;
00207
00208     typedef struct {
00209
00210         #define SL_MQTT_SRVR_NETCONN
00211         _IP6 0x01
00212         #define SL_MQTT_SRVR_NETCONN
00213         _URL 0x02
00214         #define SL_MQTT_SRVR_NETCONN
00215         _SEC 0x04
00216         _u32          netconn_flags;
00217         _const char    *server
00218         _addr;
00219         _u16          port_number;
00220         _u8           method;
00221         _u32          cipher;
00222         _u32          n_files;
00223         char * _const   *secure
00224         _files; /*SL needs 4 files*/
00225
00226     } SlMqttServerParams_t;
00227
00228     typedef struct
00229     {
00230         SlMqttServerParams_t server
00231         _info;
00232         _u16  loopback_port;
00233         _u32  rx_tsk_priority;
00234         _u32  resp_time;
00235         bool aux_debug_en;
00236         _i32 (*dbg_print)(_const char
00237             *pcFormat, ...);
00238     } SlMqttServerCfg_t;
00239

```

```
00244     _i32 sl_ExtLib_MqttServerInit(_const
00245         SLMqttServerCfg_t *cfg,
00246                                         _cons
00247         t SLMqttServerCbs_t *cbs);
00248
00249     _i32 sl_ExtLib_MqttTopicEnroll(_cons
00250         t char *topic);
00251
00252     static inline _i32 sl_ExtLib_MqttTop
00253         icSub(_const char *topic)
00254         {
00255             return sl_ExtLib_MqttTopicEn
00256             roll(topic);
00257         }
00258
00259     _i32 sl_ExtLib_MqttTopicDisenroll(_c
00260         onst char *topic);
00261
00262     static inline _i32 sl_ExtLib_MqttTop
00263         icUnsub(_const char *topic)
00264         {
00265             return sl_ExtLib_MqttTopicDi
00266             senroll(topic);
00267         }
00268
00269     _i32 sl_ExtLib_MqttServerSend(_const
00270         char *topic, _const void *data, _i32 len,
00271                                         _u8 q
00272         os, bool retain, _u32 flags);
00273 /* End SL MQTT Server API */
00274
00275 #ifdef __cplusplus
00276 }
00277 #endif
00278
00279
00280
00281
```

```
00309 #endif // __SL_MQTT_SRVR_H__
00310
00311
```



Generated on Thu Jan 15 2015 18:28:55 for sl_mqtt_server by
[doxygen](#) 1.8.0

sl_mqtt_server

Main Page	Modules	Classes	Files
Class List	Class Index	Class Members	

SIMqttServerCbs_t Member List

This is the complete list of members for **SIMqttServerCbs_t**, including all inherited members.

sl_ExtLib_MqttConn	SIMqttServerCbs_t
sl_ExtLib_MqttDisconnect	SIMqttServerCbs_t
sl_ExtLib_MqttEvent	SIMqttServerCbs_t
sl_ExtLib_MqttRecv	SIMqttServerCbs_t

Generated on Thu Jan 15 2015 18:28:55 for sl_mqtt_server by
[doxygen](#) 1.8.0

sl_mqtt_server

Main Page	Modules	Classes	Files
Class List	Class Index	Class Members	

SIMqttServerParams_t Member List

This is the complete list of members for [SIMqttServerParams_t](#), including all inherited members.

cipher	SIMqttServerParams_t
method	SIMqttServerParams_t
n_files	SIMqttServerParams_t
netconn_flags	SIMqttServerParams_t
port_number	SIMqttServerParams_t
secure_files	SIMqttServerParams_t
server_addr	SIMqttServerParams_t

Generated on Thu Jan 15 2015 18:28:55 for sl_mqtt_server by
[doxygen](#) 1.8.0

sl_mqtt_server

Main Page	Modules	Classes	Files
Class List	Class Index	Class Members	

SIMqttServerCfg_t Member List

This is the complete list of members for **SIMqttServerCfg_t**, including all inherited members.

<code>aux_debug_en</code>	<code>SIMqttServerCfg_t</code>
<code>dbg_print</code>	<code>SIMqttServerCfg_t</code>
<code>loopback_port</code>	<code>SIMqttServerCfg_t</code>
<code>resp_time</code>	<code>SIMqttServerCfg_t</code>
<code>rx_tsk_priority</code>	<code>SIMqttServerCfg_t</code>
<code>server_info</code>	<code>SIMqttServerCfg_t</code>

Generated on Thu Jan 15 2015 18:28:55 for sl_mqtt_server by
[doxygen](#) 1.8.0