eXosip2 configuration API

General purpose API.

Data Structures

struct eXosip_dns_cache
struct eXosip_tls_credentials_s
struct eXosip_tls_ctx_s

Macros

#define EXOSIP_OPT_UDP_KEEP_ALIVE (EXOSIP_OPT_BASE_OPTION+1)
#define EXOSIP_OPT_AUTO_MASQUERADE_CONTACT (EXOSIP_OPT_BASE_OPTION+2)
#define EXOSIP_OPT_USE_RPORT (EXOSIP_OPT_BASE_OPTION+7)
#define EXOSIP_OPT_SET_IPV4_FOR_GATEWAY (EXOSIP_OPT_BASE_OPTION+8)
#define EXOSIP_OPT_ADD_DNS_CACHE (EXOSIP_OPT_BASE_OPTION+9)
#define EXOSIP_OPT_DELETE_DNS_CACHE (EXOSIP_OPT_BASE_OPTION+10)
#define EXOSIP_OPT_SET_IPV6_FOR_GATEWAY (EXOSIP_OPT_BASE_OPTION+12)
#define EXOSIP_OPT_ADD_ACCOUNT_INFO (EXOSIP_OPT_BASE_OPTION+13)
#define EXOSIP_OPT_DNS_CAPABILITIES (EXOSIP_OPT_BASE_OPTION+14)
#define EXOSIP_OPT_SET_DSCP (EXOSIP_OPT_BASE_OPTION+15)
#define EXOSIP_OPT_REGISTER_WITH_DATE (EXOSIP_OPT_BASE_OPTION+16)
#define EXOSIP_OPT_SET_HEADER_USER_AGENT (EXOSIP_OPT_BASE_OPTION+17)
#define EXOSIP_OPT_ENABLE_DNS_CACHE (EXOSIP_OPT_BASE_OPTION+18)
#define EXOSIP_OPT_ENABLE_AUTOANSWERBYE (EXOSIP_OPT_BASE_OPTION+19)
#define EXOSIP_OPT_ENABLE_IPV6 (EXOSIP_OPT_BASE_OPTION+20)
#define EXOSIP_OPT_ENABLE_REUSE_TCP_PORT (EXOSIP_OPT_BASE_OPTION+21)
#define EXOSIP_OPT_SET_TLS_VERIFY_CERTIFICATE (EXOSIP_OPT_BASE_OPTION+500)
#define EXOSIP_OPT_SET_TLS_CERTIFICATES_INFO (EXOSIP_OPT_BASE_OPTION+501)
#define EXOSIP_OPT_SET_TLS_CLIENT_CERTIFICATE_NAME (EXOSIP_OPT_BASE_OPTION+502)
#define EXOSIP_OPT_SET_TLS_SERVER_CERTIFICATE_NAME (EXOSIP_OPT_BASE_OPTION+503)
#define EXOSIP_OPT_SET_TSC_SERVER (EXOSIP_OPT_BASE_OPTION+1001)

Enumerations

enum eXosip_tls_ctx_error {
  TLS_OK = 0,
  TLS_ERR_NO_RAND = -1,
  TLS_ERR_NO_DH_PARAM = -2,
  TLS_ERR_NO_PW = -3,
  TLS_ERR_NO_ROOT_CA = -4,
  TLS_ERR_MISSING_AUTH_PART = -5
}

Functions

struct eXosip_t * eXosip_malloc (void)
int eXosip_init (struct eXosip_t *excontext)
void eXosip_quit (struct eXosip_t *excontext)
int eXosip_lock (struct eXosip_t *excontext)

int	**eXosip_unlock** (struct eXosip_t *excontext)
int	**eXosip_execute** (struct eXosip_t *excontext)
int	**eXosip_set_option** (struct eXosip_t *excontext, int opt, const void *value)
struct osip_naptr *	**eXosip_dnsutils_naptr** (struct eXosip_t *excontext, const char *domain, const char *protocol, const char *transport, int keep_in_cache)
int	**eXosip_dnsutils_dns_process** (struct osip_naptr *output_record, int force)
int	**eXosip_dnsutils_rotate_srv** (struct osip_srv_record *output_record)
int	**eXosip_listen_addr** (struct eXosip_t *excontext, int transport, const char *addr, int port, int family, int secure)
int	**eXosip_reset_transports** (struct eXosip_t *excontext)
int	**eXosip_set_socket** (struct eXosip_t *excontext, int transport, int socket, int port)
void	**eXosip_set_user_agent** (struct eXosip_t *excontext, const char *user_agent)
const char *	**eXosip_get_version** (void)
int	**eXosip_set_cbsip_message** (struct eXosip_t *excontext, CbSipCallback cbsipCallback)
void	**eXosip_enable_ipv6** (int ipv6_enable)
void	**eXosip_masquerade_contact** (struct eXosip_t *excontext, const char *public_address, int port)
int	**eXosip_find_free_port** (struct eXosip_t *excontext, int free_port, int transport)

Detailed Description

Macro Definition Documentation

#define EXOSIP_OPT_UDP_KEEP_ALIVE (EXOSIP_OPT_BASE_OPTION+1)

int *: interval for keep alive packets (UDP, TCP, TLS, DTLS)

#define EXOSIP_OPT_AUTO_MASQUERADE_CONTACT (EXOSIP_OPT_BASE_OPTION+2)

int *: specific re-usage of "rport"

#define EXOSIP_OPT_USE_RPORT (EXOSIP_OPT_BASE_OPTION+7)

int *: enable or disable rport in via

#define EXOSIP_OPT_SET_IPV4_FOR_GATEWAY (EXOSIP_OPT_BASE_OPTION+8)

char *: usually, this is the proxy address

#define EXOSIP_OPT_ADD_DNS_CACHE (EXOSIP_OPT_BASE_OPTION+9)

struct **eXosip_dns_cache** *: force some cache entry to avoid DNS

#define EXOSIP_OPT_DELETE_DNS_CACHE (EXOSIP_OPT_BASE_OPTION+10)

struct **eXosip_dns_cache** *: force removal of some cache entry to avoid DNS

#define EXOSIP_OPT_SET_IPV6_FOR_GATEWAY (EXOSIP_OPT_BASE_OPTION+12)

char *: usually, this is the proxy address

#define EXOSIP_OPT_ADD_ACCOUNT_INFO (EXOSIP_OPT_BASE_OPTION+13)

struct eXosip_account_info *: internal stuff

#define EXOSIP_OPT_DNS_CAPABILITIES (EXOSIP_OPT_BASE_OPTION+14)

int *: 0 to disable, 2 to use NAPTR/SRV record

#define EXOSIP_OPT_SET_DSCP (EXOSIP_OPT_BASE_OPTION+15)

int *: set a dscp value for SIP socket

#define EXOSIP_OPT_REGISTER_WITH_DATE (EXOSIP_OPT_BASE_OPTION+16)

int *: enable usage of Date header in REGISTER

#define EXOSIP_OPT_SET_HEADER_USER_AGENT (EXOSIP_OPT_BASE_OPTION+17)

char *: set the User-Agent header

#define EXOSIP_OPT_ENABLE_DNS_CACHE (EXOSIP_OPT_BASE_OPTION+18)

int *: 0 to disable use of cache

#define EXOSIP_OPT_ENABLE_AUTOANSWERBYE (EXOSIP_OPT_BASE_OPTION+19)

int *: 0 to disable automatic answer of BYE

#define EXOSIP_OPT_ENABLE_IPV6 (EXOSIP_OPT_BASE_OPTION+20)

int *: 0 to disable -this is a per-eXosip_t parameter for using IPv6 DNS request

#define EXOSIP_OPT_ENABLE_REUSE_TCP_PORT (EXOSIP_OPT_BASE_OPTION+21)

int *: 0 to disable, 1 to enable reusing local tcp port for outgoing tcp connection

#define EXOSIP_OPT_SET_TLS_VERIFY_CERTIFICATE (EXOSIP_OPT_BASE_OPTION+500)

int *: enable verification of certificate for TLS connection

#define EXOSIP_OPT_SET_TLS_CERTIFICATES_INFO (EXOSIP_OPT_BASE_OPTION+501)

eXosip_tls_ctx_t *: client and/or server certificate/ca-root/key info

#define EXOSIP_OPT_SET_TLS_CLIENT_CERTIFICATE_NAME (EXOSIP_OPT_BASE_OPTION+502)

char*: user can choose a specific certifcate present in Windows Certificate Store

#define EXOSIP_OPT_SET_TLS_SERVER_CERTIFICATE_NAME (EXOSIP_OPT_BASE_OPTION+503)

char*: user can choose a specific certifcate present in Windows Certificate Store

#define EXOSIP_OPT_SET_TSC_SERVER (EXOSIP_OPT_BASE_OPTION+1001)

void*: set the tsc tunnel handle

Enumeration Type Documentation

enum eXosip_tls_ctx_error

An enumeration which describes the error which can occur while setting the eXosip_tls_ctx

Enumerator:

TLS_OK	yippieh, everything is fine :)
TLS_ERR_NO_RAND	no absolute path to the random file was specified
TLS_ERR_NO_DH_PARAM	no absolute path to the diifie hellman file was specified
TLS_ERR_NO_PW	no password was specified
TLS_ERR_NO_ROOT_CA	no absolute path to the rootCA file was specified
TLS_ERR_MISSING_AUTH_PART	something is missing: the private key or the certificate

Function Documentation

struct eXosip_t* eXosip_malloc (void) read

Allocate an eXosip context.

Returns

a new allocated eXosip_t instance.

int eXosip_init (struct eXosip_t * excontext)

Initiate the eXtented oSIP library.

Parameters

excontext eXosip_t instance.

void eXosip_quit (struct eXosip_t * excontext)

Release ressource used by the eXtented oSIP library.

Parameters

excontext eXosip_t instance.

int eXosip_lock (struct eXosip_t * excontext)

Lock the eXtented oSIP library.

Parameters

excontext eXosip_t instance.

int eXosip_unlock (struct eXosip_t * excontext)

UnLock the eXtented oSIP library.

Parameters

excontext eXosip_t instance.

int eXosip_execute (struct eXosip_t * excontext)

Process (non-threaded mode ONLY) eXosip events.

Parameters

excontext eXosip_t instance.

int eXosip_set_option (struct eXosip_t * excontext,
int opt,
const void * value
)

Set eXosip options. See eXosip_option for available options.

Parameters

excontext eXosip_t instance.
opt option to configure.
value value for options.

read

struct osip_naptr*
eXosip_dnsutils_naptr (struct eXosip_t * excontext,
const char * domain,
const char * protocol,
const char * transport,

```
                                int                 keep_in_cache
                              )
```

Start and return osip_naptr context. Note that DNS results might not yet be available.

Parameters

excontext	eXosip_t instance.
domain	domain name for NAPTR record
protocol	protocol to use ("SIP")
transport	transport to use ("UDP")
keep_in_cache	keep result in cache if >0

```
int eXosip_dnsutils_dns_process ( struct osip_naptr *  output_record,
                                  int                  force
                                )
```

Continue to process asynchronous DNS request (if implemented).

Parameters

output_record	result structure.
force	force waiting for final answer if >0

```
int
eXosip_dnsutils_rotate_srv ( struct osip_srv_record *  output_record )
```

Rotate first SRV entry to last SRV entry.

Parameters

output_record	result structure.

```
int eXosip_listen_addr ( struct eXosip_t *  excontext,
                         int                transport,
                         const char *       addr,
                         int                port,
                         int                family,
                         int                secure
                       )
```

Listen on a specified socket.

Parameters

excontext	eXosip_t instance.
transport	IPPROTO_UDP for udp. (soon to come: TCP/TLS?)
addr	the address to bind (NULL for all interface)
port	the listening port. (0 for random port)

family the IP family (AF_INET or AF_INET6).
secure 0 for UDP or TCP, 1 for TLS (with TCP).

int eXosip_reset_transports (struct eXosip_t * excontext)

Reset transport sockets.

Parameters
excontext eXosip_t instance.

int eXosip_set_socket (struct eXosip_t * excontext, int transport, int socket, int port)

Listen on a specified socket.

Parameters
excontext eXosip_t instance.
transport IPPROTO_UDP for udp. (soon to come: TCP/TLS?)
socket socket to use for listening to UDP sip messages.
port the listening port for masquerading.

void eXosip_set_user_agent (struct eXosip_t * excontext, const char * user_agent)

Set the SIP User-Agent: header string.

Parameters
excontext eXosip_t instance.
user_agent the User-Agent header to insert in messages.

const char* eXosip_get_version (void)

Get the eXosip version as a sring

int eXosip_set_cbsip_message (struct eXosip_t * excontext, CbSipCallback cbsipCallback)

Set a callback to get sent and received SIP messages.

Parameters

excontext eXosip_t instance.

cbsipCallback the callback to retreive messages.

void eXosip_enable_ipv6 (int ipv6_enable)

Use IPv6 instead of IPv4. (global setting)

DEPRECATED: you MUST use EXOSIP_OPT_ENABLE_IPV6 to configure each eXosip_t independantly.

THIS CODE DOES NOTHING, REPLACE WITH

eXosip_set_option(excontext, EXOSIP_OPT_ENABLE_IPV6, &val);

Parameters

ipv6_enable This paramter should be set to 1 to enable IPv6 mode.

void eXosip_masquerade_contact (struct eXosip_t * excontext,
const char * public_address,
int port
)

This method is used to replace contact address with the public address of your NAT. The ip address should be retreived manually (fixed IP address) or with STUN. This address will only be used when the remote correspondant appears to be on an DIFFERENT LAN.

Parameters

excontext eXosip_t instance.

public_address the ip address.

port the port for masquerading.

If set to NULL, then the local ip address will be guessed automatically (returns to default mode).

int eXosip_find_free_port (struct eXosip_t * excontext,
int free_port,
int transport
)

This method is used to find out an free IPPROTO_UDP or IPPROTO_TCP port.

Parameters

excontext eXosip_t instance.

free_port initial port for search.

transport IPPROTO_UDP or IPPROTO_TCP protocol.