

How-To send or update registrations.

The eXtented eXosip stack

Initiate a registration

To start a registration, you need to build a default REGISTER request by providing several mandatory headers.

You can start as many registration you want even in one eXosip_t context.

```
osip_message_t *reg = NULL;
int rid;
int i;

eXosip_lock (ctx);
rid = eXosip_register_build_initial_register (ctx, "sip:me@sip.antisip.com", "sip:sip.antisip.com",
      NULL, 1800, &reg);
if (rid < 0)
{
    eXosip_unlock (ctx);
    return -1;
}

osip_message_set_supported (reg, "100rel");
osip_message_set_supported (reg, "path");

i = eXosip_register_send_register (ctx, rid, reg);
eXosip_unlock (ctx);
return i;
```

The returned element of eXosip_register_build_initial_register is the registration identifier that you can use to update your registration. In future events about this registration, you'll see that registration identifier when applicable.

Contact headers in REGISTER

You should let eXosip2 manage contact headers alone. The setup you have provided will generate various behavior, all being controlled by eXosip2. See the "NAT and Contact header" section in setup documentation.

Set password(s)!

Usually, you will need a login/password to access a service!

eXosip can be used to access several service at the same time and thus the realm (identify the service) needs to be provided if you are using several services in the same instance.

The simple way when you use one service with username being the same as the login:

```
eXosip_lock (ctx);
eXosip_add_authentication_info (ctx, login, login, passwd, NULL, NULL);
eXosip_unlock (ctx);
```

OR the complete way:

```
eXosip_lock (ctx);
```

```
eXosip_add_authentication_info (ctx, login, login, passwd, NULL, "sip.antisip.com");
eXosip_add_authentication_info (ctx, from_username, login, passwd, NULL, "otherservice.com");
eXosip_unlock (ctx);
```

Delete all registration

This feature is not advised by sip specification, but it exists for some purpose & reasons! You can send "*" in Contact header of a REGISTER to ask for immediate removal of all active registrations for a particular SIP agent:

```
rid = eXosip_register_build_initial_register (ctx, "sip:me@sip.antisip.com", "sip:sip.antisip.com",
      "*", 1800, &reg);
```

Update a registration

You just need to reuse the registration identifier:

```
int i;

eXosip_lock (ctx);
i = eXosip_register_build_register (ctx, rid, 1800, &reg);
if (i < 0)
{
    eXosip_unlock (ctx);
    return -1;
}

eXosip_register_send_register (ctx, rid, reg);
eXosip_unlock (ctx);
```

Note: The above code also shows that the stack is sometimes able to build and send a default SIP messages with only one API call

Closing the registration

A softphone should delete its registration on the SIP server when terminating. To do so, you have to send a REGISTER request with the expires header set to value "0".

The same code as for updating a registration is used with 0 instead of 1800 for the expiration delay.

```
int i;

eXosip_lock (ctx);
i = eXosip_register_build_register (ctx, rid, 0, &reg);
if (i < 0)
{
    eXosip_unlock (ctx);
    return -1;
}

eXosip_register_send_register (ctx, rid, reg);
eXosip_unlock (ctx);
```

Discard registration context

If you need to delete a context without sending a REGISTER with expires 0, you can use `eXosip_register_remove` to release memory.

```
int i;  
  
eXosip_lock (ctx);  
i = eXosip_register_remove (ctx, rid);  
if (i == 0) {  
}  
eXosip_unlock (ctx);
```