

NAME

sasl_client_new - Cyrus SASL documentation

SYNOPSIS

```
#include <sasl/sasl.h>
```

```
int sasl_client_new(const char *service,  
                  const char *serverFQDN,  
                  const char *iplocalport,  
                  const char *ipremoteport,  
                  const sasl_callback_t *prompt_supp,  
                  unsigned flags,  
                  sasl_conn_t ** pconn);
```

DESCRIPTION

```
int sasl_client_new(const char *service,
```

```
const char *serverFQDN,
```

```
const char *iplocalport,
```

```
const char *ipremoteport,
```

```
const sasl_callback_t *prompt_supp,
```

```
unsigned flags,
```

```
sasl_conn_t ** pconn);
```

sasl_client_new() creates a new SASL context. This context will be used for all SASL calls for one connection. It handles both authentication and integrity/encryption layers after authentication.

Parameters

- ⊕ **service** - the registered name of the service (usually the protocol name) using SASL (e.g. "imap").
- ⊕ **serverFQDN** - the fully qualified domain name of the server (e.g. "serverhost.example.com").

- ⊕ **iplocalport** - the IP and port of the local side of the connection, or NULL. If iplocalport is NULL it will disable mechanisms that require IP address information. This string must be in one of the following formats: "a.b.c.d;port" (IPv4), "e:f:g:h:i:j:k:l;port" (IPv6), or "e:f:g:h:i:j:a.b.c.d;port" (IPv6)
- ⊕ **ipremoteport** - the IP and port of the remote side of the connection, or NULL (see iplocalport)
- ⊕ **prompt_supp** - a list of client interactions supported that is unique to this connection. If this parameter is NULL the global callbacks (specified in sasl_client_init(3)) will be used. See sasl_callbacks(3) for more information.
- ⊕ **flags** - are connection flags (see below)
- ⊕ **pconn** - the connection context allocated by the library. This structure will be used for all future SASL calls for this connection.

Connection Flags

Flags that may be passed to **sasl_client_new()**:

- ⊕ **SASL_SUCCESS_DATA**: The protocol supports a server-last send

⊕

SASL_NEED_PROXY: Force the use of a mechanism that supports an authorization id that is not the authentication id.

RETURN VALUE

SASL callback functions should return SASL return codes. See sasl.h for a complete list. **SASL_OK** indicates success.

The following return codes indicate errors and should either be handled or the authentication session should be quit:

- ⊕ **SASL_NOMECH**: No mechanism meets requested properties
- ⊕ **SASL_BADPARAM**: Error in config file
- ⊕ **SASL_NOMEM**: Not enough memory to complete operation

SEE ALSO

RFC 4422,:saslmn:*sasl(3)*, *sasl_callbacks(3)*, *sasl_client_init(3)*, *sasl_client_start(3)*,
sasl_client_step(3), *sasl_setprop(3)*

AUTHOR

The Cyrus Team

COPYRIGHT

1993-2016, The Cyrus Team