# Library file to access Device Token Repo server

This library is used by master server.

## Methods

DeviceTokenRepo.createSession(dev\_id, dev\_serial, dev\_usage, dev\_type, group\_id, nonce, secured, skey, iv)

Calls Device Token Repo to create a new session.

### Steps

1. Get TOKEN\_REPO\_URL from config file (parameter tokenRepo.url)
2. Get masterId and masterSecret from config file (parameter tokenRepo.masterId and tokenRepo.masterSecret)
3. Call <http://TOKEN_REPO_URL/session/create> with masterId, masterSecret, dev\_id, dev\_serial, dev\_usage, dev\_type, group\_id, nonce, secured, skey and iv
4. Return response
5. Calls are POS and JSON

DeviceTokenRepo.getSession(session\_id, dev\_serial)

Calls Device Token Repo to get a session.

### Steps

1. Get TOKEN\_REPO\_URL from config file (parameter tokenRepo.url)
2. Get masterId and masterSecret from config file (parameter tokenRepo.masterId and tokenRepo.masterSecret)
3. Call <http://TOKEN_REPO_URL/session/get> with masterId, masterSecret, session\_id and dev\_serial
4. Return response
5. Calls are POS and JSON

DeviceTokenRepo.deleteSession(session\_id)

Calls Device Token Repo to delete a session.

### Steps

1. Get TOKEN\_REPO\_URL from config file (parameter tokenRepo.url)
2. Get masterId and masterSecret from config file (parameter tokenRepo.masterId and tokenRepo.masterSecret)
3. Call <http://TOKEN_REPO_URL/session/delete> with masterId, masterSecret and session\_id
4. Return response
5. Calls are POS and JSON

DeviceTokenRepo.createToken(dev\_id, dev\_serial, dev\_usage, dev\_type, group\_id, secured, skey, iv, expires)

Calls Device Token Repo to create a new Token.

### Steps

1. Get TOKEN\_REPO\_URL from config file (parameter tokenRepo.url)
2. Get masterId and masterSecret from config file (parameter tokenRepo.masterId and tokenRepo.masterSecret)
3. Call <http://TOKEN_REPO_URL/token/create> with masterId, masterSecret, dev\_id, dev\_serial, dev\_usage, dev\_type, group\_id, secured, skey, iv and expires
4. Return response
5. Calls are POS and JSON

DeviceTokenRepo.deleteToken(access\_token, refresh\_token)

Calls Device Token Repo to delete a token.

### Steps

1. Get TOKEN\_REPO\_URL from config file (parameter tokenRepo.url)
2. Get masterId and masterSecret from config file (parameter tokenRepo.masterId and tokenRepo.masterSecret)
3. Call <http://TOKEN_REPO_URL/token/delete> with masterId, masterSecret, access\_token and refresh\_token
4. Return response
5. Calls are POS and JSON

DeviceTokenRepo.deleteAllTokens(dev\_id)

Calls Device Token Repo to delete all tokens of device.

### Steps

1. Get TOKEN\_REPO\_URL from config file (parameter tokenRepo.url)
2. Get masterId and masterSecret from config file (parameter tokenRepo.masterId and tokenRepo.masterSecret)
3. Call <http://TOKEN_REPO_URL/token/deleteAll> with masterId, masterSecret and dev\_id
4. Return response
5. Calls are POS and JSON

DeviceTokenRepo.refresh(dev\_serial, refresh\_token)

Calls Device Token Repo to refresh a token.

### Steps

1. Get TOKEN\_REPO\_URL from config file (parameter tokenRepo.url)
2. Get masterId and masterSecret from config file (parameter tokenRepo.masterId and tokenRepo.masterSecret)
3. Call <http://TOKEN_REPO_URL/token/refresh> with masterId, masterSecret, refresh\_token and dev\_id
4. Return response
5. Calls are POS and JSON

DeviceTokenRepo.deleteAllTokens(dev\_id)

Calls Device Token Repo to delete all tokens of device.

### Steps

1. Get TOKEN\_REPO\_URL from config file (parameter tokenRepo.url)
2. Get masterId and masterSecret from config file (parameter tokenRepo.masterId and tokenRepo.masterSecret)
3. Call <http://TOKEN_REPO_URL/token/deleteAll> with masterId, masterSecret and dev\_id
4. Return response
5. Calls are POS and JSON

DeviceTokenRepo.renewSessionKey(access\_token, payload, new\_skey, new\_iv)

Calls Device Token Repo to renew a session key.

### Steps

1. Get TOKEN\_REPO\_URL from config file (parameter tokenRepo.url)
2. Get masterId and masterSecret from config file (parameter tokenRepo.masterId and tokenRepo.masterSecret)
3. Call [http://TOKEN\_REPO\_URL/token/renewSessionKey](http://TOKEN_REPO_URL/token/renewSessionKey%20) with masterId, masterSecret, access\_token, payload, new\_skey and new\_iv
4. Return response
5. Calls are POS and JSON