

Mxnet2 API

1. サーバへの接続

rInit_MatrixAPI でサーバとのセッションを開始します。ログインしないと他の API を呼び出すことはできません。戻り値が ≥ 0 で成功。

プログラムの最初の rInit_MatrixAPI の呼出でサーバにログイン(セッション開始)します。その後、何度呼び出しても既存セッションを使い回します。

2. サーバとの接続解放

rRelease_MatrixNet でサーバとのセッションを終了します。

必ず rInit_MatrixAPI と同じ回数呼び出さなければなりません。rInit_MatrixAPI を複数回呼び出していれば、同じ回数 rRelease_MatrixNet を呼び出します。最後の rRelease_MatrixNet 呼び出しでセッションが終了します。

3. サーバ側でのセッション解放

SessionTimeout で指定した間隔以内に MxNet2 API を呼び出さないとサーバ側でセッションを一方的に解放してしまいます。セッションが解放された状態で API を呼び出すとエラー(—105)が返ります。rInit_MatrixAPI で再接続して続行できます。

4. リモートライセンス管理の方法

16個ある AppSlot のいずれかに最大ライセンス数をセットします。例えば、同時に10つのプログラム起動を許可するには、AppSlot に 10 を書き込みます。ここでは AppSlot5 に書き込んだものとします

プログラムの最初のインスタンスが MxNet2 サーバに接続後、ユーザコード、AppSlot を指定して rLogin_MatrixNet API を呼び出すと 9 が返ります

```
short r = rLogin_MatrixNet( usercode, 5, 1 ) // r==9
```

2番目のインスタンスが呼び出すと8が返ります

```
short r = rLogin_MatrixNet( usercode, 5, 1 ) // r==8
```

10番目のインスタンスが呼び出すと 0 が返ります

```
short r = rLogin_MatrixNet( usercode, 5, 1 ) // r==0
```

11番目のインスタンスが呼び出すと-31(ライセンス空き無し)が返ります

```
short r = rLogin_MatrixNet( usercode, 5, 1 ) // r==-31
```

インスタンス起動時に $r < 0$ であればプログラムを起動しない/ライセンス無状態で実行するようにします。

ライセンス更新

ライセンスを取得したインスタンスは、ライセンスを保持するために定期的に rLogin_MatrixNet を呼び出し続けなければなりません(=ライセンス更新)。10個のインスタンスがライセンス更新をしている間は、Login は0を返します

```
short r = rLogin_MatrixNet( usercode, 5, 1 ) // r==0
```

10個のインスタンスのいずれかがプログラム異常のためライセンス更新をしなければ、サーバ側でそのインスタンスのライセンスを解放、Login は1が返すようになります。

```
short r = rLogin_MatrixNet( usercode, 5, 1 ) // r==1
```

この時 11番目のインスタンスが LogIn 呼び出すと 0 が返り、このインスタンスがライセンスを取得します。ライセンスを失ったインスタンスが LogInを呼び出すと -31 が返り、ライセンスを取得できません。

インスタンスは、rLogOut_MatrixNet でライセンスを解放します。10個のインスタンスのいずれかが LogOut を呼び出すと1(ライセンスが1つ空いている)が返ります

```
short r = rLogOut_MatrixNet( usercode, 5, 1 ) // r==1
```

サーバとの接続を解放(rRelease_MatrixAPI)すると、明示的にライセンスを解放しなくても、保持しているライセンスは自動解放されます。

ライセンス方式 : Session vs IP

これまでの例では、インスタンス(Session)単位でライセンスが増減しました。複数のインスタンスが同一コンピュータ上で実行していても、それぞれのインスタンスがライセンスを1つずつ取得します。

IP (コンピュータ)単位でライセンスを取得することも可能です。このためには、MxNet2 オプションのライセンスを IP に設定します。

- 同一コンピュータで複数のインスタンスがライセンスを取得しても、最初のインスタンスでライセンスが1つ減少しますが、2番目以降のインスタンスがライセンスを取得しても減少しません。
- 同一コンピュータでライセンスを保持している複数のインスタンスの1つがライセンスを解放しても、ライセンスは減少しません。最後のインスタンスが解放した時点でライセンスは1つ減少します。

同一コンピュータ(IP)からの同一 AppSlot に対するライセンス要求は、複数のインスタンスからあっても、1つのインスタンスからあったようになります。

LoginTimeOut(ログインタイムアウト)の他に、SessionTimeOut(セッションタイムアウト)があります。クライアントがサーバに接続後、セッションタイムアウトで指定した間隔以内に、APIを発行しないとクライアントが居なくなったと見なして、サーバは一方的に接続を解放します。サーバに接続していないと API 呼び出しは失敗します。

APPSLOT へのライセンス数の設定

ライセンス管理を行うには、指定 AppSlot にライセンス数が書き込まれていなければなりません。Web インターフェースの”管理画面2” のドングル設定で設定してください。

API 呼び出しについて

MxNet2 API は非同期ではありません。サーバに接続できないとブロックします。メインスレッドでブロックすると、プログラムが固まった状態になります。

これを避けるためすべてのサンプルプログラムでは MxNet2 API を別スレッドで呼び出しています。

安定したネットワーク環境で常に接続できるのであれば、別スレッドで呼び出す必要はありません。