

DBInterface 가

Last update: 2023/05/11

www.softbase.co.kr
Copyright © SOFTBase Inc. All rights reserved.

DBInterface 가 1
..... 1
..... 1

DBInterface 가

xFrameDBInterface

(xFrameDBInterface.ocx)

: /XPLUS/xplus_dbinterface

- [xplus_dbinterface.xml](#)
- [xplus_dbinterface.js](#)
- [xplus_dbinterface.js](#)

```
// DB /
function btnConnect_on_mouseup(objInst)
{
    //
    if (ctrlDB.innerctrl.IsConnected() == false) {
        fnOpenDatabae();
    }
    // DB가 ,
    else {
        // DB
        ctrlDB.innerctrl.Disconnect();
    }

    ShowOpenDatabaeStatus();
}

// DB
function ShowAutoCommitStatus()
{
    if (ctrlDB.innerctrl.IsConnected()) {
        if (ctrlDB.innerctrl.GetAutoCommit() == 1) {
            chkAutoCommit.setcheck(true);
            btnCommit.setenable(false);
            btnRollback.setenable(false);
        }
        else {
            chkAutoCommit.setcheck(false);
            btnCommit.setenable(true);
            btnRollback.setenable(true);
        }
    }
}
```

```
    }
  }
}

// DB
function ShowOpenDatabaeStatus()
{
  if (ctrlDB.innerctrl.IsConnected()) {
    // DB          DB
    btnConnect.setText("Disconnect");
    pnlConnect.setenable(true);
  }
  else {
    // DB          DB
    btnConnect.setText("Connect");
    pnlConnect.setenable(true);
  }

  ShowAutoCommitStatus();
}

/**
 * DB
 */
function fnOpenDatabae()
{
  var nDBKind = cbDBKind.getSelectedcode();
  var strDBName = fieldDBname.gettext();
  var strServiceName = fieldServiceName.gettext();
  var strIPAddress = fieldIPAddress.gettext();
  var nPort = Number(fieldPort.gettext());
  var strUID = fieldUID.gettext();
  var strPWD = fieldPWD.gettext();

  if (ctrlDB.innerctrl.IsConnected() == true) {
    return true;
  }

  var bConnect = ctrlDB.innerctrl.Connect(nDBKind, strDBName,
strServiceName, strIPAddress, nPort, strUID, strPWD);

  return bConnect;
}

//
function btnExecQuery_on_mouseup(objInst)
{
  var ret, strQuery, nFieldCount, nFieldIndex, nRecordCount, strValue,
nRowIndex, nColumnIndex;

  //
```

```
gridResult.deleteall();
gridResult.deleteallcolumn();

if (ctrlDB.innerctrl.IsConnected() == false && fnOpenDatabae() == false)
{
    screen.alert("DB가          .");
    return;
}

strQuery = mmQuery.gettext();
if (strQuery.length <= 0) {
    screen.alert("          .");
    return;
}

//
if (ctrlDB.innerctrl.ExecuteSQL(strQuery) == false) {
    screen.alert("          .");
    return;
}

//
nRecordCount = ctrlDB.innerctrl.GetResultRecordCount();
factory.consoleprint("nRecordCount : " + nRecordCount);
if (nRecordCount <= 0) {
    screen.alert("          = " + nRecordCount);
    return;
}

//          (Select 가          ,          가 0 )
nFieldCount = ctrlDB.innerctrl.GetResultFieldCount();
if (nFieldCount <= 0) {
    screen.alert("          = " + nRecordCount);
    return;
}

factory.consoleprint("nFieldCount : " + nFieldCount);

//
for (nFieldIndex = 0; nFieldIndex < nFieldCount; nFieldIndex++) {
    nColumnIndex = gridResult.addcolumn();
    if (nColumnIndex < 0) {
        screen.alert("          가          .");
        continue;
    }

    //
    gridResult.setheadertext(0, nColumnIndex,
ctrlDB.innerctrl.getResultFieldName(nFieldIndex));

    gridResult.setColumnAutoWidth(nColumnIndex, false);
    gridResult.setColumnWordEllipsis(nColumnIndex, true);
}
```

```
}

//
nResult = ctrlDB.innerctrl.GetResultRecordFetch();
factory.consoleprint("Result record fetch : " + nResult);

//          가          Loop
while (nResult == 1) {
    //          가
    nRowIndex = gridResult.additem(false, false);

    //
    for(nColumnIndex = 0; nColumnIndex < nFieldCount; nColumnIndex++) {
        strValue = ctrlDB.innerctrl.GetFieldValue(nColumnIndex);
        gridResult.setitemtext(nRowIndex, nColumnIndex, strValue);
    }

    //
    nResult = ctrlDB.innerctrl.GetResultRecordFetch();
}
}

// Commit
function btnCommit_on_mouseup(objInst)
{
    if (ctrlDB.innerctrl.IsConnected() == false) { return; }

    if (ctrlDB.innerctrl.Commit() == false) { screen.alert("Commit error"); }
}

// Rollback
function btnRollback_on_mouseup(objInst)
{
    if (ctrlDB.innerctrl.IsConnected() == false) { return; }

    if (ctrlDB.innerctrl.Rollback() == false) { screen.alert("Rollback
error"); }
}

// AutoCommit
function chkAutoCommit_on_click(objInst)
{
    if (chkAutoCommit.getcheck() == true) {
        if (ctrlDB.innerctrl.GetAutoCommit() == 0) {
            ctrlDB.innerctrl.SetAutoCommit(true);
        }
    }
    else {
        if (ctrlDB.innerctrl.GetAutoCommit() == 1) {
            ctrlDB.innerctrl.SetAutoCommit(false);
        }
    }
}
```

```
}  
  ShowAutoCommitStatus();  
}
```

From:

<https://technet.softbase.co.kr/wiki/> - **xFrame5 TechNet**

Permanent link:

https://technet.softbase.co.kr/wiki/guide/xplus/xplus_dbinterface



Last update: **2023/05/11 16:21**