XPlusHttpSvr 가이드

Last update: 2023/05/11

www.softbase.co.kr

목차

XPlusHttpSvr 가이드

이 화면은 XPlusHttpSvr.ocx 컴포넌트에 대한 샘플 화면이다.

XPlusHttpSvr.ocx 컴포넌트는 HTTP 프로토콜에 대한 서버 기능을 제공하는 컴포넌트이다.

예시

템플릿 위치: /XPLUS/xplus_httpsvr

템플릿 파일

- xplus_httpsvr.xml
- xplus_httpsvr.js
- 새창으로 실행

화면 스크립트

```
////////////////////////////////////////////////////////////////////////////
/////////////
// EVENT START
////////////////////////////////////////////////////////////////////////////
/////////////

/**
 * XTcpSvrComm 데이터 수신 이벤트 처리
 * @param objInst XTcpSvrComm 오브젝트
 * @param pIpAddr 데이터를 송신한 시스템의 IP 주소
 * @param nPortNo 데이터를 송신한 시스템의 TCP 포트 번호
 * @param nSocketKey 원격지 시스템 연결 소켓 키 값
 * @param pHeader HTTP 헤더 문자열(맨 마지막 \r\n은 제거되어 있음)
 * @param nLength 수신한 데이터 길이
 * @param pTranKey 수신한 데이터에 대한 Key값, GetData 함수에서 사용됨
 */
function
objXPlusHttpSvr_OnReceive(objInst,pRemoteIpAddr,nRemotePortNo,nSocketKey,pHe
ader,nLength,pDataKey)
{
    factory.consoleprint("OnReceive> pRemoteIpAddr = " + pRemoteIpAddr);
    factory.consoleprint("OnReceive> nRemotePortNo = " + nRemotePortNo);
    factory.consoleprint("OnReceive> nSocketKey = " + nSocketKey);
    factory.consoleprint("OnReceive> pHeader = " + pHeader);
    factory.consoleprint("OnReceive> nLength = " + nLength);
    factory.consoleprint("OnReceive> pDataKey = " + pDataKey);
```

```
    fldRecvIpAddr.settext(pRemoteIpAddr);
    fldRecvPortNo.settext(nRemotePortNo);
    fldRecvDataLength.settext(nLength);

    /**
   * 실제 수신된 데이터를 구함
     * @return strRecvData 수신한 데이터
     */
    var strRecvData = objXPlusHttpSvr.innerctrl.GetData(pDataKey);
    factory.consoleprint("RecvData = " + strRecvData);
    fldRecvData.settext(strRecvData);

    if (chkAutoReply.getcheck()) {
        SendResponseData(pRemoteIpAddr, nRemotePortNo, nSocketKey);
    }

    return;
}

/**
 * TCP 세션이 해제되면 발생하는 이벤트 처리
 * CloseSession 함수 호출시에는 이벤트가 발생하지 않는다.
 * @param objInst XTcpSvrComm 오브젝트
 * @param pRemoteIpAddr 세션이 해제된 시스템의 IP 주소
 * @param nRemotePortNo 세션이 해제된 시스템의 TCP 포트 번호
 * @param nSocketKey 원격지 시스템 연결 소켓 키 값
 */
function
objXPlusHttpSvr_OnClose(objInst,pRemoteIpAddr,nRemotePortNo,nSocketKey)
{
    factory.consoleprint("OnClose> pRemoteIpAddr = " + pRemoteIpAddr);
    factory.consoleprint("OnClose> nRemotePortNo = " + nRemotePortNo);
    factory.consoleprint("OnClose> nSocketKey = " + nSocketKey);

    var nRowCount = grdSessionList.getrowcount();
    var i;

    for(i = 0; i < nRowCount; i++) {
        if(grdSessionList.getitemtext(i, 0) == pRemoteIpAddr) {
            if(grdSessionList.getitemtext(i, 1) == nRemotePortNo) {
                grdSessionList.deleterow(i);
                break;
            }
        }
    }

    return;
}

/**
 * TCP 세션이 연결되면 발생하는 이벤트 처리
```

```
 * @param objInst XTcpSvrComm 오브젝트
 * @param pRemoteIpAddr 원격지 시스템의 IP 주소
 * @param nRemotePortNo 원격지 시스템의 TCP 포트 번호
 * @param nSocketKey 원격지 시스템 연결 소켓 키 값
 */
function
objXPlusHttpSvr_OnConnect(objInst,pRemoteIpAddr,nRemotePortNo,nSocketKey)
{
    factory.consoleprint("OnConnect> pRemoteIpAddr = " + pRemoteIpAddr);
    factory.consoleprint("OnConnect> nRemotePortNo = " + nRemotePortNo);
    factory.consoleprint("OnConnect> nSocketKey = " + nSocketKey);

    var nRow = grdSessionList.additem();
    grdSessionList.setitemtext(nRow, 0, pRemoteIpAddr);
    grdSessionList.setitemtext(nRow, 1, nRemotePortNo);
    grdSessionList.setitemtext(nRow, 2, nSocketKey);
}

////////////////////////////////////////////////////////////////////////////
/////////////
// EVENT END
////////////////////////////////////////////////////////////////////////////
/////////////

// XTcpSvrComm 초기화
function btnInitXSvrComm_on_mouseup(objInst)
{
    var strLogDir = "C:\\xFrame\\log";
    var nRet;

    /**
     * XPlusHttpSvr 컴포넌트를 초기화한다.
     * @param pLogDir 로그 저장 디렉토리
     * @param pLogFilePrefix 로그 파일 이름
     * @param pLogLevel 로그 레벨("DEBUG"<"INFO"<"WARN"<"ERROR")
     *                  ("DEBUG"로 지정시에만 송수신 데이터 덤프 로그가 기록됨)
     * @param nLogEncFlag 로그 파일 내용 암호화 여부
     * @param nBindPortNo Listen TCP 포트 번호
     * @return
     *    0 : Success
     *    1 : Invalid Length Field Length
     *    9 : Already Initialized
     */
    nRet = objXPlusHttpSvr.innerctrl.InitXPlusHttpSvr(strLogDir,
"XPlusHttpSvr", "DEBUG", 0, fldBindPortNo.gettext());
    factory.consoleprint("InitXPlusHttpSvr Return Value = " + nRet);
    if(nRet != 0) {
        screen.alert("InitXPlusHttpSvr() Fail, ErrorCode = " + nRet);
    }

    /**
```

```
     * TCP 송수신 데이터에 대한 UTF8 문자셋 처리 설정
     * @param nSocketKey 소켓 키 값
     * @returns 연결 여부 값
     *        0: Success
     *        1: Invalid Parameter
     */
    nRet = objXPlusHttpSvr.innerctrl.SetUtf8Flag(1);
    factory.consoleprint("SetUtf8Flag Return Value = " + nRet);

    /**
     * TCP 데이터 송신 타임아웃(단위: 밀리초) 설정
     * @param nSocketKey 소켓 키 값
     * @returns 연결 여부 값
     *        0: Success
     *        1: Invalid Parameter
     */
    nRet = objXPlusHttpSvr.innerctrl.SetSendTimeout(5000);    // 5초
    factory.consoleprint("SetSendTimeout Return Value = " + nRet);
}

// XTcpSvrComm 시작 버튼 클릭 이벤트 처리
function btnStartXSvrComm_on_mouseup(objInst)
{
    /**
     * XPlusHttpSvr 소켓을 열고 TCP 접속 대기
    * @param nBindPortNo Listen할 TCP 포트 번호 (0 값을 지정할 경우,
InitXPlusHttpSvr 함수에서 지정한 포트 사용)
     * @returns 처리 결과 값
     *        0: Success
     *        1: Invalid Port No
     *        2: Fail To Create Socket
     *        3: Fail To Socket Listen
     *        9: Not Initialized
     */
    nRet = objXPlusHttpSvr.innerctrl.StartXPlusHttpSvr(0);
    factory.consoleprint("StartXSvrComm StartXPlusHttpSvr Value = " + nRet);
    if(nRet != 0) {
        screen.alert("StartXPlusHttpSvr() Fail, ErrorCode = " + nRet);
    }
}

// XTcpSvrComm 종료 버튼 클릭 이벤트 처리
function btnStopXSvrComm_on_mouseup(objInst)
{
    /**
     * XPlusHttpSvr 소켓을 닫고, 연결된 TCP 접속 해제
     * @param nCloseSessionFlag 연결되어 있는 TCP 세션 해제 여부
     * @param nFireEventFlag 세션 해제 이벤트 발생 처리 여부 플래그
     * @returns 처리 결과 값
     *        0: Success
     *        9: Not Initialized
```

```
     */
    var nCloseSessionFlag = 1;
    var nFireEventFlag = 0;

    nRet = objXPusHttpSvr.innerctrl.StopXPlusHttpSvr(nCloseSessionFlag,
nFireEventFlag);
    factory.consoleprint("StopXPlusHttpSvr Return Value = " + nRet);
    if(nRet != 0) {
        screen.alert("StopXPlusHttpSvr() Fail, ErrorCode = " + nRet);
    }

    if(nCloseSessionFlag == 1 && nFireEventFlag == 0) {
        grdSessionList.deleteall();
    }
}

// 세션 해제 버튼 클릭 이벤트 처리
function btnCloseSession_on_mouseup(objInst)
{
    var nCheckedRowCount = grdSessionList.getcheckedrowcount();
    if(nCheckedRowCount == 0) {
        screen.alert("연결 해제할 세션을 체크하세요.");
        return;
    }

    var nCheckedRow = grdSessionList.getcheckedrow(0);
    var strRemoteIpAddr = grdSessionList.getitemtext(nCheckedRow, 0);
    var strRemotePortNo = grdSessionList.getitemtext(nCheckedRow, 1);
    var nRet;

    /**
   * 지정된 TCP 세션을 해재한다.
     * @param strRemoteIpAddr 데이터 수신 대상 시스템 IP 주소
     * @param strRemotePortNo 데이터 수신 대상 포트 번호
     * @param nFireEventFlag 세션 해제 이벤트 발생 처리 여부 플래그
     * @return
     *    0 : OK
     *    1 : Invalid Parameter
     *    2 : Fail To Find Session
     *    9 : Not Initialized
     */
    var nFireEventFlag = 1;
    nRet = objXPusHttpSvr.innerctrl.CloseSession(strRemoteIpAddr,
strRemotePortNo, nFireEventFlag);
    // nRet = objXPusHttpSvr.innerctrl.CloseSessionByKey(nSocketKey,
nFireEventFlag);
    factory.consoleprint("CloseSession Return Value = " + nRet);
    if(nRet != 0) {
        screen.alert("CloseSession() Fail, ErrorCode = " + nRet);
    }
}
```

```
// 데이터 송신 버튼 클릭 이벤트 처리
function btnSendData_on_mouseup(objInst)
{
    var nCheckedRowCount = grdSessionList.getcheckedrowcount();
    if(nCheckedRowCount == 0) {
        screen.alert("데이터를 송신할 세션을 체크하세요.");
        return;
    }

    var nCheckedRow = grdSessionList.getcheckedrow(0);
    var strRemoteIpAddr = grdSessionList.getitemtext(nCheckedRow, 0);
    var nRemotePortNo = grdSessionList.getitemtext(nCheckedRow, 1);
    var nSocketKey = grdSessionList.getitemtext(nCheckedRow, 2);
    var strSendData = fldSendData.gettext();
    var nRet;

    SendResponseData(strRemoteIpAddr, nRemotePortNo, nSocketKey,
strSendData, strSendData);
}

function SendResponseData(strRemoteIpAddr, nRemotePortNo, nSocketKey,
strData)
{
    var strSendDataArr = [];

    if (strData === undefined || strData.length == 0) {
        strData = factory.getsystemtime("%Y-%M-%D %h:%m:%s %ms");
    }

    /**
   * 데이터를 지정된 세션으로 송신한다.
     * @param strRemoteIpAddr 데이터 수신 대상 시스템 IP 주소
     * @param strRemotePortNo 데이터 수신 대상 포트 번호
     * @param nSendDataLength 송신 데이터 길이 (단위: 바이트)
     * @param strSendData 송신 데이터
     * @return
     *    0 : OK
     *    1 : Fail To Find Session
     *    2 : Invalid Parameter
     *    3 : Session is Not Connected
     *    4 : Fail To Send Length Part
     *    5 : Fail To Send Data Part
     *    9 : Not Initialized
     */

    // HTTP Header 데이터는 맨 뒤에 \r\n 데이터를 붙여야 함(예:
"aaa:bbb\r\nccc:ddd\r\n");
    var strHeader = "";

    nRet = objXPlusHttpSvr.innerctrl.SendData(strRemoteIpAddr,
nRemotePortNo, strHeader, strData);
```

```
    // nRet = objXPlusHttpSvr.innerctrl.SendDataByKey(nSocketKey, "",
strData);
    factory.consoleprint("SendData Return Value = " + nRet);
    if(nRet != 0) {
        screen.alert("SendData() Fail, ErrorCode = " + nRet);
    }
}

// 수신 데이터 정보 초기화 버튼 클릭 이벤트 처리
function btnClearData_on_mouseup(objInst)
{
    fldRecvIpAddr.settext("");
    fldRecvPortNo.settext("");
    fldRecvDataLength.settext("");
    fldRecvData.settext("");
}

function btnGetSessionCount_on_mouseup(objInst)
{
    screen.alert(objXPlusHttpSvr.innerctrl.GetSessionCount());
}
```