XPlusWebSockSvr 가이드

Last update: 2024/10/29

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

목차

XPlusWebSockSvr 가이드 1
예시 1
화면 스크립트 1

XPlusWebSockSvr 가이드

이 화면은 XPlusWebSockSvr.ocx 컴포넌트 샘플 화면이다.

XPlusWebSockSvr.ocx 컴포넌트는 Web Socket 프로토콜에 대한 서버 기능을 제공하는 컴포넌트이다.

WebSocket 클라이언트용 xFrame5 샘플 화면은 /HTML5/COMPONENT/WEBSOCKET/websocket 템플릿 화면을 참조한다.

WebSocket 클라이언트용 순수 HTML 샘플은 툴 설치 디렉토리/template/XPLUS/websocket 폴더의 websocket_client.html 파일을 참조한다.

예시

템플릿 위치: /XPLUS/xplus_websocksvr

템플릿 파일

- xplus_websocksvr.xml
- xplus_websocksvr.js
- 새창으로 실행

화면 스크립트

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

/**
 * 웹소켓 서버 데이터 수신 이벤트 처리
  * @param objInst XPlusWebSockSvr 오브젝트
  * @param pRemoteIpAddr 데이터를 송신한 시스템의 IP 주소
  * @param nRemotePortNo 데이터를 송신한 시스템의 TCP 포트 번호
  * @param nSocketKey 원격지 시스템 연결 소켓 키 값
  * @param nLength 수신한 데이터 길이
  * @param pDataKey 수신한 데이터에 대한 Key값, GetData 함수에서 사용됨
  */
function objXPlusWebSockSvr_OnReceive(bjInst, pRemoteIpAddr, nRemotePortNo,
nSocketKey, nLength, pDataKey)
{
    var strRecvData;

    factory.consoleprint("OnReceive> pRemoteIpAddr = " + pRemoteIpAddr);
    factory.consoleprint("OnReceive> nRemotePortNo = " + nRemotePortNo);
```

```
    factory.consoleprint("OnReceive> nSocketKey = " + nSocketKey);
    factory.consoleprint("OnReceive> nLength = " + nLength);
    factory.consoleprint("OnReceive> pDataKey = " + pDataKey);

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

    /**
   * 실제 수신된 데이터를 구함
     * @return strRecvData 수신한 데이터
     */
    strRecvData = this.objXPlusWebSockSvr.innerctrl.GetData(pDataKey);

    // screen.alert(strRecvData);
    // factory.consoleprint("RecvData = " + strRecvData);
    // fldRecvData.settext(strRecvData);

  // 수신 데이터가 100바이트가 넘는 경우, 일부 데이터만 잘라서 표시
    if (strRecvData.length > 1000) {
        // fldRecvData.settext(strRecvData.substring(strRecvData.length -
20));
        this.fldRecvData.settext(strRecvData.substr(0, 1000));
    }
    else {
        this.fldRecvData.settext(strRecvData);
    }

    // ECHO 모드인 경우, 수신한 데이터를 그대로 리턴
    if (this.chkEchoMode.getcheck()) {
        /**
     * 데이터를 지정된 세션으로 송신한다.
         * @param strRemoteIpAddr 데이터 수신 대상 시스템 IP 주소
         * @param strRemotePortNo 데이터 수신 대상 포트 번호
         * @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
         */
        nRet = this.objXPlusWebSockSvr.innerctrl.SendData(pRemoteIpAddr,
nRemotePortNo, strRecvData);
        // nRet = objXPlusWebSockSvr.innerctrl.SendDataByKey(nSocketKey, "",
strSendData);

        factory.consoleprint("SendData Return Value = " + nRet);
        if (nRet != 0) {
```

```
            screen.alert("SendData() Fail, ErrorCode = " + nRet);
        }
    }

    return;
}

/**
 * 웹소켓 세션이 해제되면 발생하는 이벤트 처리
 * CloseSession 함수 호출시에는 이벤트가 발생하지 않는다.
 * @param objInst XPlusWebSockSvr 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 objXPlusWebSockSvr_OnClose(objInst, pRemoteIpAddr, nRemotePortNo,
nSocketKey)
{
    var i, nRowCount;

    factory.consoleprint("OnClose> pRemoteIpAddr = " + pRemoteIpAddr);
    factory.consoleprint("OnClose> nRemotePortNo = " + nRemotePortNo);
    factory.consoleprint("OnClose> nSocketKey = " + nSocketKey);

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

    return;
}

/**
 * 웹소켓 세션이 연결되면 발생하는 이벤트 처리
 * @param objInst XPlusWebSockSvr 오브젝트
 * @param pRemoteIpAddr 원격지 시스템의 IP 주소
 * @param nRemotePortNo 원격지 시스템의 TCP 포트 번호
 * @param nSocketKey 원격지 시스템 연결 소켓 키 값
 */
function objXPlusWebSockSvr_OnConnect(objInst, pRemoteIpAddr, nRemotePortNo,
nSocketKey)
{
    var nRow;

    factory.consoleprint("OnConnect> pRemoteIpAddr = " + pRemoteIpAddr);
    factory.consoleprint("OnConnect> nRemotePortNo = " + nRemotePortNo);
```

```
    factory.consoleprint("OnConnect> nSocketKey = " + nSocketKey);

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

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

// XPlusWebSockSvr 초기화
function btnInitXSvrComm_on_mouseup(objInst)
{
    var nRet, strLogDir, strLogLevel;

  // 로그 파일 저장 디렉토리 절대 경로
    strLogDir = "C:\\xFrame5\\log";
    strLogLevel = "DEBUG";        // "DEBUG"로 지정시에만 송수신 데이터 덤프 로그가
기록됨

    /**
     * XPlusWebSockSvr 컴포넌트를 초기화한다. 초기에 한번만 호출하면 된다.
     * @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 = this.objXPlusWebSockSvr.innerctrl.InitXPlusWebSockSvr(strLogDir,
"XPlusWebSockSvr", strLogLevel, 0, this.fldBindPortNo.gettext());
    factory.consoleprint("InitXPlusWebSockSvr Return Value = " + nRet);
    if (nRet != 0) {
        screen.alert("InitXPlusWebSockSvr() Fail, ErrorCode = " + nRet);
    }

    /**
   * 데이터 송신 타임아웃(단위: 밀리초) 설정
     * @param nSocketKey 소켓 키 값
     * @returns 연결 여부 값
     *     0 : Success
     *     1 : Invalid Parameter
     */
    nRet = this.objXPlusWebSockSvr.innerctrl.SetSendTimeout(5000);    // 5초
```

```
    factory.consoleprint("SetSendTimeout Return Value = " + nRet);
}

// XPlusWebSockSvr 시작 버튼 클릭 이벤트 처리
function btnStartXSvrComm_on_mouseup(objInst)
{
    var nRet;

    /**
   * 웹소켓 포트를 열고 웹 소켓 접속 대기
    * @param nBindPortNo Listen할 TCP 포트 번호 (0 값을 지정할 경우,
InitXPlusWebSockSvr 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 = this.objXPlusWebSockSvr.innerctrl.StartXPlusWebSockSvr(0);
    factory.consoleprint("StartXPlusWebSockSvr Value = " + nRet);
    if (nRet != 0) {
        screen.alert("StartXPlusWebSockSvr() Fail, ErrorCode = " + nRet);
    }
}

// XPlusWebSockSvr 종료 버튼 클릭 이벤트 처리
function btnStopXSvrComm_on_mouseup(objInst)
{
    var nCloseSessionFlag, nFireEventFlag;

    /**
     * XPlusWebSockSvr 소켓을 닫고, 연결된 WebSocket 세션 해제
     * @param nCloseSessionFlag 연결되어 있는 WebSocket 세션 해제 여부
     * @param nFireEventFlag 세션 해제 이벤트 발생 처리 여부 플래그
     * @returns 처리 결과 값
     *     0 : Success
     *     9 : Not Initialized
     */
    nCloseSessionFlag = 1;
    nFireEventFlag = 0;

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

    if (nCloseSessionFlag == 1 && nFireEventFlag == 0) {
```

```
        this.grdSessionList.deleteall();
    }
}

// 세션 해제 버튼 클릭 이벤트 처리
function btnCloseSession_on_mouseup(objInst)
{
    var nRet, nCheckedRowCount, nCheckedRow, strRemoteIpAddr,
strRemotePortNo, nFireEventFlag;

    nCheckedRowCount = this.grdSessionList.getcheckedrowcount();
    if (nCheckedRowCount == 0) {
        screen.alert("연결 해제할 세션을 체크하세요.");
        return;
    }

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

    /**
   * 지정된 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
     */
    nFireEventFlag = 1;

    nRet = this.objXPlusWebSockSvr.innerctrl.CloseSession(strRemoteIpAddr,
strRemotePortNo, nFireEventFlag);
    // nRet = objXPlusWebSockSvr.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 nRet, nCheckedRowCount, nCheckedRow, strRemoteIpAddr, nRemotePortNo,
nSocketKey, strSendData;

    nCheckedRowCount = this.grdSessionList.getcheckedrowcount();
```

```
    if (nCheckedRowCount == 0) {
        screen.alert("데이터를 송신할 세션을 체크하세요.");
        return;
    }

    nCheckedRow = this.grdSessionList.getcheckedrow(0);
    strRemoteIpAddr = this.grdSessionList.getitemtext(nCheckedRow, 0);
    nRemotePortNo = this.grdSessionList.getitemtext(nCheckedRow, 1);
    nSocketKey = this.grdSessionList.getitemtext(nCheckedRow, 2);

    strSendData = this.fldSendData.gettext();

    /**
   * 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
     */
    nRet = this.objXPlusWebSockSvr.innerctrl.SendData(strRemoteIpAddr,
nRemotePortNo, strSendData);
    // nRet = objXPlusWebSockSvr.innerctrl.SendDataByKey(nSocketKey,
strSendData);

    factory.consoleprint("SendData Return Value = " + nRet);
    if (nRet != 0) {
        screen.alert("SendData() Fail, ErrorCode = " + nRet);
    }
}

function btnSendPing_on_mouseup(objInst)
{
    var nRet, nCheckedRowCount, nCheckedRow, strRemoteIpAddr, nRemotePortNo,
nSocketKey, strSendData;

    nCheckedRowCount = this.grdSessionList.getcheckedrowcount();
    if (nCheckedRowCount == 0) {
        screen.alert("데이터를 송신할 세션을 체크하세요.");
        return;
    }

    nCheckedRow = this.grdSessionList.getcheckedrow(0);
    strRemoteIpAddr = this.grdSessionList.getitemtext(nCheckedRow, 0);
```

```
    nRemotePortNo = this.grdSessionList.getitemtext(nCheckedRow, 1);
    nSocketKey = this.grdSessionList.getitemtext(nCheckedRow, 2);

    strSendData = this.fldSendData.gettext();

    /**
     * PING 데이터를 지정된 세션으로 송신한다.
     * @param strRemoteIpAddr 데이터 수신 대상 시스템 IP 주소
     * @param strRemotePortNo 데이터 수신 대상 포트 번호
     * @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
     */
    nRet = this.objXPlusWebSockSvr.innerctrl.SendPing(strRemoteIpAddr,
nRemotePortNo);
    // nRet = objXPlusWebSockSvr.innerctrl.SendPingByKey(nSocketKey);

    factory.consoleprint("SendPing Return Value = " + nRet);
    if (nRet != 0) {
        screen.alert("SendPing() Fail, ErrorCode = " + nRet);
    }
}

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

// "세션 갯수" 버튼 이벤트
function btnGetSessionCount_on_mouseup(objInst)
{
    screen.alert(this.objXPlusWebSockSvr.innerctrl.GetSessionCount());
}
```

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

Permanent link:
https://technet.softbase.co.kr/wiki/guide/xplus/xplus_websocksvr

Last update: **2024/10/29 18:15**