

Last update: 2023/10/31

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

.....	1
.....	1
.....	1

Promise async, await (ECMAScript (ECMA-262))

API async function , await API

API IE

API domodal, getscreentype

on_popupload, on_popupdestroy가

: /HTML5/SCREEN/SYNC/sync_domodal

- [sync_domodal.xml](#)
- [sync_domodal.js](#)
- [sync_domodal.css](#)

```
var SCREEN_BASE_DIR = "/HTML5/SCREEN/SYNC/";

//
function screen_on_load()
{
    factory.showconsoletrace(true);
}

/**
 * factory.loadpopup API 가
 *
 * @param popup_screen
 * @param popup_name factory.loadpopup API
 */
async function screen_on_popupload(popup_screen, popup_name)
{
    var log_prefix, popup_ret;
```

```

log_prefix = "mainscreen:screen_on_popupload> ";

factory.consoleprint(log_prefix + "start");

factory.consoleprint(log_prefix + "popup_name = " + popup_name);
if (factory.isobject(popup_screen)) {
    factory.consoleprint(log_prefix + "screen = " +
popup_screen.getscreenurl());
}

// domodal
if (this.chk_calldomodal.getcheck()) {
    factory.consoleprint(log_prefix + "screen type = " +
popup_screen.getscreentype());
    factory.consoleprint(log_prefix + "before WaitPopupUnload");
    popup_ret = await WaitPopupUnload(screen, popup_name,
popup_screen.getscreentype() == 0 ? false : true);
    factory.consoleprint(log_prefix + "after WaitPopupUnload");
    factory.consoleprint(log_prefix + "popup_ret = " + popup_ret);
}

factory.consoleprint(log_prefix + "end");
}

/**
 *          unloadpopup API
 *
 * @param popup_screen
 * @param popup_name factory.loadpopup API
 * @param result          unloadpopup API
 */
function screen_on_popupdestroy(popup_screen, popup_name, result)
{
    var log_prefix;

    log_prefix = "mainscreen:screen_on_popupdestroy> ";

    factory.consoleprint(log_prefix + "start");
    factory.consoleprint(log_prefix + "popup_name = " + popup_name);
    factory.consoleprint(log_prefix + "result = " + result);
    factory.consoleprint(log_prefix + "end");
}

// "loadpopup"
function btn_loadpopup_on_mouseup(objInst)
{
    var log_prefix, ret, is_modal;

    log_prefix = "mainscreen:loadpopup> ";

    is_modal = this.chk_modalpopup.getcheck();

```

```

//          /
factory.consoleprint(log_prefix + "before loadpopup");
ret = factory.loadpopupex("POPUP_SCREEN_ASYNC",
    SCREEN_BASE_DIR + "business_screen", "POPUP SCREEN ASYNC",
    false, XFD_BORDER_RESIZE, 0, 0, 200, 500, true, true, is_modal,
    screen, "POPUPSCREEN_EXTRADATA");
factory.consoleprint(log_prefix + "after loadpopup");
factory.consoleprint(log_prefix + "ret = " + ret);
}

// "loadpopupsync"
async function btn_loadpopupsync_on_mouseup(objInst)
{
    var log_prefix, popup_screen, ret, popup_ret, is_modal;

    log_prefix = "mainscreen:loadpopupsync_modal> ";

    is_modal = this.chk_modalpopup.getcheck();

//          /
factory.consoleprint(log_prefix + "before loadpopupsync");
ret = await factory.loadpopupsync("POPUP_SCREEN_SYNC",
    SCREEN_BASE_DIR + "business_screen", "POPUP SCREEN SYNC",
    false, XFD_BORDER_RESIZE, 0, 0, 200, 500, true, true, is_modal,
    screen, "POPUPSCREEN_EXTRADATA");
factory.consoleprint(log_prefix + "after loadpopupsync");
factory.consoleprint(log_prefix + "return = " + ret);

//          URL
popup_screen = factory.findpopup("POPUP_SCREEN_SYNC");
if (factory.isobject(popup_screen)) {
    factory.consoleprint(log_prefix + "screen = " +
popup_screen.getscreenurl());
}

// domodal
if (this.chk_calldomodal.getcheck()) {
    factory.consoleprint(log_prefix + "before WaitPopupUnload");
    popup_ret = await WaitPopupUnload(screen, "POPUP_SCREEN_SYNC",
true);
    factory.consoleprint(log_prefix + "after WaitPopupUnload");

    return popup_ret;
}

return ret;
}

// "loadportletpopup"
function btn_loadportletpopup_on_mouseup(objInst)
{

```

```
var log_prefix, ret;

log_prefix = "mainscreen:loadportletpopup> ";

//
factory.consoleprint(log_prefix + "before loadportletpopup");
ret = screen.loadportletpopup("POPUP_PORTLET_ASYNC",
    SCREEN_BASE_DIR + "business_screen", "POPUP PORTLET ASYNC",
    false, XFD_BORDER_RESIZE, 0, 0, 200, 500, true, true, false,
    "POPUPSCREEN_EXTRADATA");
factory.consoleprint(log_prefix + "after loadportletpopup");
factory.consoleprint(log_prefix + "ret = " + ret);
}

// "loadportletpopupsync"
async function btn_loadportletpopupsync_on_mouseup(objInst)
{
    var log_prefix, popup_screen, ret, popup_ret;

    log_prefix = "mainscreen:loadportletpopupsync> ";

    //
    factory.consoleprint(log_prefix + "before loadportletpopupsync");
    ret = await screen.loadportletpopupsync("POPUP_PORTLET_SYNC",
        SCREEN_BASE_DIR + "business_screen", "POPUPPORTLET SYNC",
        false, XFD_BORDER_RESIZE, 0, 0, 200, 500, true, true, false,
        "POPUPSCREEN_EXTRADATA");
    factory.consoleprint(log_prefix + "after loadportletpopupsync");
    factory.consoleprint(log_prefix + "return = " + ret);

    //
    // URL
    popup_screen = screen.findportletpopup("POPUP_PORTLET_SYNC");
    if (factory.isobject(popup_screen)) {
        factory.consoleprint(log_prefix + "screen = " +
popup_screen.getscreenurl());
    }

    // domodal
    if (this.chk_calldomodal.getcheck()) {
        factory.consoleprint(log_prefix + "before WaitPopupUnload");
        popup_ret = await WaitPopupUnload(screen, "POPUP_PORTLET_SYNC",
true);
        factory.consoleprint(log_prefix + "after WaitPopupUnload");
        factory.consoleprint(log_prefix + "popup_ret = " + popup_ret);

        return popup_ret;
    }

    return ret;
}
}
```

```
/**
 *
 *
 * @param is_modal
 * @param obj_screen
 * @param popup_name
 * @param is_portlet_popup
 *
 * @returns          unloadpopup API
 */
async function WaitPopupUnload(obj_screen, popup_name, is_portlet_popup)
{
    var log_prefix, obj_extra_data, popup_screen;

    log_prefix = "mainscreen:WaitPopupUnload> ";

    //
    if (is_portlet_popup) {
        popup_screen = obj_screen.findportletpopup(popup_name);
    }
    else {
        popup_screen = factory.findpopup(popup_name);
    }

    if (!factory.isobject(popup_screen)) { return null; }

    //          domodal API          ,
    //          unloadpopup API
    // domodal API          "on_popupdestroy"
    // domodal API          , on_popupdestroy
    //          unloadpopup API
    factory.consoleprint(log_prefix + "before domodal");
    popup_ret = await popup_screen.domodal();
    factory.consoleprint(log_prefix + "after domodal");
    factory.consoleprint(log_prefix + "popup_ret = " + popup_ret);

    return popup_ret;
}
```

From:

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

Permanent link:

https://technet.softbase.co.kr/wiki/guide/screen/sync/sync_domodal



Last update: **2023/10/31 10:27**