Not quite what you are looking for? You may want to try:

- GIS data visualization of VisualBasic hybrids with SVG/CSS
- Internet of Things Security Architecture

highlights off

Custom JavaScript confirm and alert message boxes not using callbacks

Vladimir Misovsky, 7 Apr 2016 CPOL

★☆☆☆☆ 1.00 (2 votes)

Rate:

This article provides an example of custom confirm and alert message boxes that don’t use callbacks

Download source files and demo page - 97.2 KB

Introduction

Many JS frameworks provide custom confirm and alert message boxes. Message box creation implementation may differ from one framework to another, but there is one common part – user selection is always processed by a callback function, which makes source code too messy, especially when multiple confirm message boxes generated inside one javascript function.

This article provides an example of custom confirm and alert message boxes that don’t use callbacks and therefore don’t break execution flow.

Background

The idea is quite simple. Message box initialization function (**VM.MsgBox.Start**) stores caller function and all its arguments in a stack, and after the user made his selection message box stores the result and calls the above function again. VM.MsgBox.Start must be called at the beginning of each function generating confirm/alert message boxes.

One more function (**VM.MsgBox.Clear**) is required to clear message box stack. VM.MsgBox.Clear must be called from a function that calls the function generating confirm/alert message boxes.

Message box icon is defined by one of the following values:

VM.MsgBoxIcon = { Excl: 0, Info: 1, Ques: 2, Stop: 3, Custom: 4 }, where first four values are considered to apply to "system" icons.

Message box result is defined by one of the following values:

VM.DlgResult = { OK : 0, Cancel: 1, Retry: 2, Abort: 3, Ignore: 4, Yes: 5, No: 6, Custom: -1, Unknown: 99 }

The result returned by the first call to Confirm is always **VM.DlgResult.Unknown**.

VM.MsgBox.Confirm and **VM.MsgBox.Alert** are used to generate custom confirm and alert message boxes respectively.

Both functions take the same set of parameters:

id: message box id

msg: message to be displayed in the message box

icon: system message box icon

src: image source for a custom message box icon

fn: message box callback function used to set message box styles/colors.

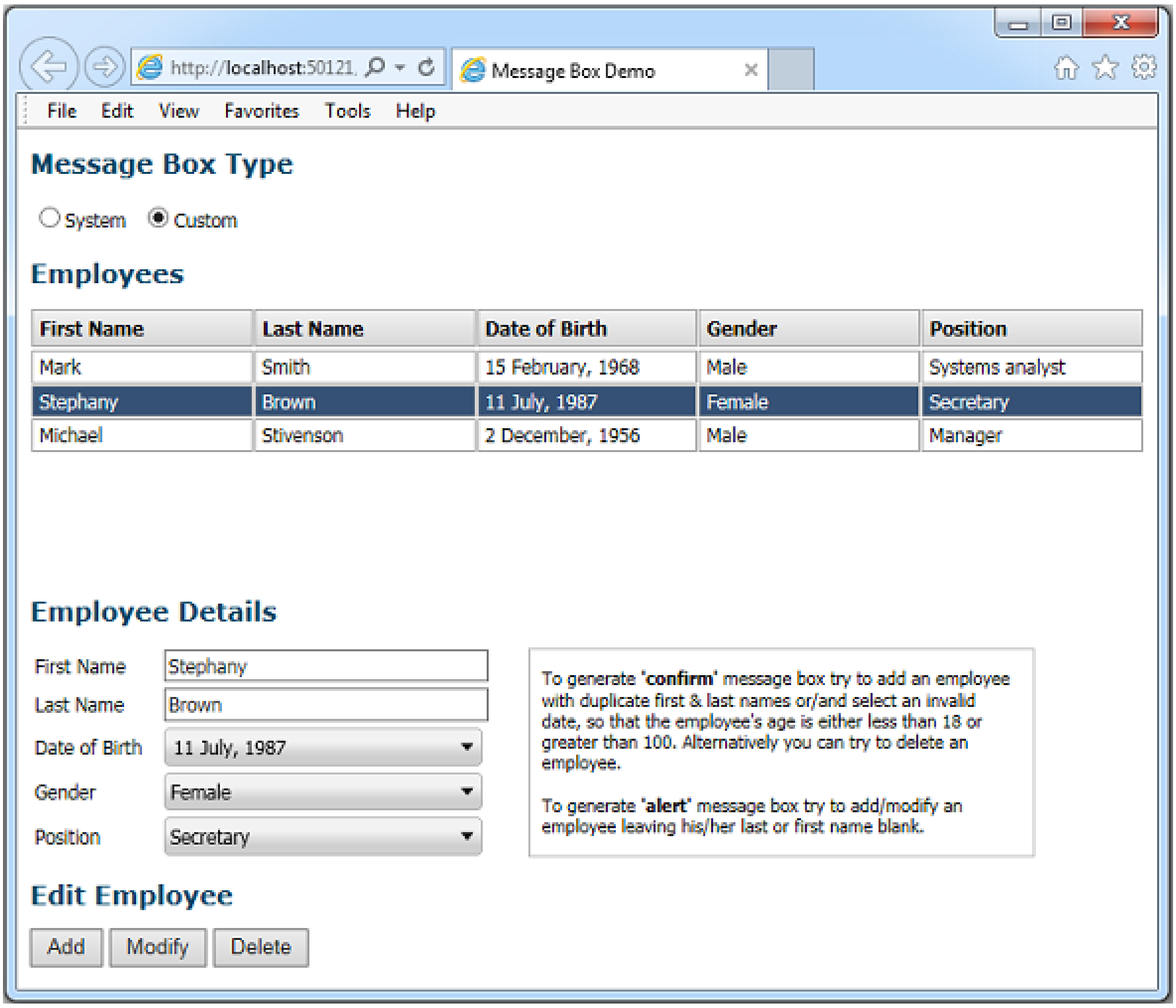
iGap: margin in pixels between message box text and the icon

vGap: vertical margin between message box components (text, check boxes, progress bars)

Last four parameters are optional. If multiple confirm/alert message boxes are generated inside one javascript function each message box must have a unique id.

Using the code

This demo project contains a single page that provides a simple Employee Editor.



Message Box Type

○ System ● Custom

Employees

First Name	Last Name	Date of Birth	Gender	Position
Mark	Smith	15 February, 1968	Male	Systems analyst
Stephany	Brown	11 July, 1987	Female	Secretary
Michael	Stivenson	2 December, 1956	Male	Manager

Employee Details

First Name	Stephany
Last Name	Brown
Date of Birth	11 July, 1987
Gender	Female
Position	Secretary

To generate '**confirm**' message box try to add an employee with duplicate first & last names or/and select an invalid date, so that the employee's age is either less than 18 or greater than 100. Alternatively you can try to delete an employee.

To generate '**alert**' message box try to add/modify an employee leaving his/her last or first name blank.

Edit Employee

Add Modify Delete

Page initialization function adds few records to the employee’s table, sets system message box icons and creates combo boxes for an employee’s date of birth, gender and position selection.

Hide Copy Code

```
function initPage()
{
    VM.SetLanguage('EN');
    VM_AddEventHandler("mainDiv", VM_EVT_BRESIZE, "OnBodyResized");
    var table = document.getElementById("employeeTbl");
    VM.SetAttr(table, "selOnClk", "1");
    VM.SetAttr(table, "clickFunc", "onRecSelected");
    // Adding few records
    addEmployee("Mark", "Smith", "1968;01;15;00;00;00", 0, 2);
    addEmployee("Stephany", "Brown", "1987;06;11;00;00;00", 1, 1);
    addEmployee("Michael", "Stivenson", "1956;11;02;00;00;00", 0, 0);
    // Initializing table adjuster (making table columns scalable & resizable)
    g_tblAdjuster = VM.TblAdjuster.Create();
    g_tblAdjuster.AddTable("employeeHdr", "employeeTbl", -1, true);
    g_tblAdjuster.AdjustTable("employeeHdr", true);
    // Setting message box icons
    VM.MsgBox.iconInfo = 'Images/mb_info.png';
    VM.MsgBox.iconExcl = 'Images/mb_excl.png';
    VM.MsgBox.iconQues = 'Images/mb_ques.png';
    VM.MsgBox.iconStop = 'Images/mb_stop.png';
    // Initializing combo boxes
    initCombos();
}
```

“Add”, “Modify” and “Delete” employee button click events are handled by onButtonClick function

Hide   Copy Code

```
function onButtonClick(m)
{
    // (m == 0) => "Add" clicked; (m == 1) => "Modify" clicked; (m == 2) => "Delete" clicked
    // Calling VM.MsgBox.Clear to clear message box stack
    VM.MsgBox.Clear();
    editEmployee(m);
}
```

editEmployee function generates “confirm-type” message boxes in the following situations

1. The full name of a new or modified employee (a combination of the last and the first name) matches an existing employee’s full name.
2. The selected date of birth is outside the allowed interval, making employee’s age either less that 18 or greater that 100 years.
3. The employee is being deleted.

The “alert-type” message box is generated if the employee’s last or first name is empty.

Hide   Shrink ▲   Copy Code

```
function editEmployee(m)
{
    // Calling VM.MsgBox.Start
    VM.MsgBox.Start();
    var v = {f: "", l: "", d: "", g: -1, p: -1};
    var res = true;
    if (m == 2)
    {
        // Confirm record deletion
        res = g_custom ? Confirm('conf_delete', g_msg[0], VM.MsgBoxIcon.Ques) : confirm(g_msg[0]);
    }
    else
    {
        v.f = document.getElementById("f_name").value;
        v.l = document.getElementById("l_name").value;
        if (v.f == '' || v.l == '')
        {
            // Either last or first name is empty
            var n = (v.f == '') ? 4 : 5;
            if (g_custom)
                Alert(g_msg[n], VM.MsgBoxIcon.Excl);
            else
                alert(g_msg[n]);
            res = false;
        }
        else if (employeeExists(m, v.f, v.l))
        {
            // Employee with the specified <last name, first name> pair already exists
            res = g_custom ? Confirm('conf_exists', g_msg[3], VM.MsgBoxIcon.Ques) :
```

```
confirm(g_msg[3]);
        }
        if (res)
        {
            for (var i=0; i<3; i++)
            {
                var divID = (i == 2) ? "pos_combo" : (i == 1) ? "gend_combo" : "dob_combo";
                var div = document.getElementById(divID);
                var combo = div ? VM.FindCtrlByParentDiv(div, true) : null;
                var list = combo ? combo.m_obj : null;
                if (list)
                {
                    if (i == 0)
                        v.d = list.GetValue();
                    else if (i == 1)
                        v.g = list.GetCurSel();
                    else
                        v.p = list.GetCurSel();
                }
            }
            var resAge = validateDate(v.d);
            if (resAge > 0)
            {
                // Improper employee's date of birth selected
                res = g_custom ? Confirm('conf_age', g_msg[resAge], VM.MsgBoxIcon.Ques) :
                      confirm(g_msg[resAge]);
            }
        }
    }
    // If res == true => proceed with editing
    if (res)
    {
        if (m == 2)
            deleteEmployee();
        else if (m == 0)
            addEmployee(v.f, v.l, v.d, v.g, v.p);
        else
            modifyEmployee(v.f, v.l, v.d, v.g, v.p);
        if (m != 1)
        {
            g_tblAdjuster.AdjustTable("employeeHdr", false);
            enableButtons();
        }
    }
}
```

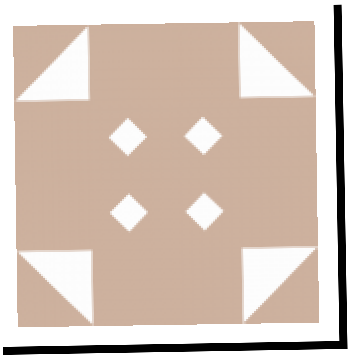
Depending on the message box type selected by the user function will either generate system or custom message boxes.

Note: Script file (VMWebControls_min.js) containing message box base classes as well as few other classes of my widgets library used in this project are supplied in a packed format.

License

Share

About the Author



Vladimir Misovsky C++ and ASP.NET web developer

Software Developer

Australia

Comments and Discussions

Add a Comment or Question ? **Search Comments** Go

First Prev Next

Screenshot of Alert Messages? new

Garry Lowther 12-Apr-16 9:41

I recommend you show us what these alert dialogue messages look like in the article?
Typically, developers will not download code unless it is worth their while.

Garry Lowther
CEO and Founder
TriSys Business Software
Cambridge, England
www.trisys.co.uk

Reply · Email · View Thread · Permalink · Bookmark

My vote of 1 new

Timothy R 8-Apr-16 16:44

This is absurdly complicated, unintuitive, and absolutely unreadable. Code should be simple, easy to read, and obvious how to use.

Promises, or clean lambdas run circles around this approach.

Reply · Email · View Thread · Permalink · Bookmark

Please don't put minimized code in your library. new

Leng Vang 8-Apr-16 14:28

It is already complex as is, don't make it worst.

Reply · View Thread · Permalink · Bookmark

My vote of 1 new

lejuif 8-Apr-16 7:29

It's far too complex to setup.
Code readability is bad.

I understand that you want to get rid of the callback hell but there are better ways to do it.
You should take a look at "promises"

Reply · Email · View Thread · Permalink · Bookmark

Refresh 1

General News Suggestion Question Bug Answer Joke Praise Rant Admin

Permalink | Advertise | Privacy | Terms of Use | Mobile
Web02 | 2.8.160919.1 | Last Updated 7 Apr 2016

Select Language ▼
Layout: fixed | fluid