# Office AMS: JavaSCRIPT injection

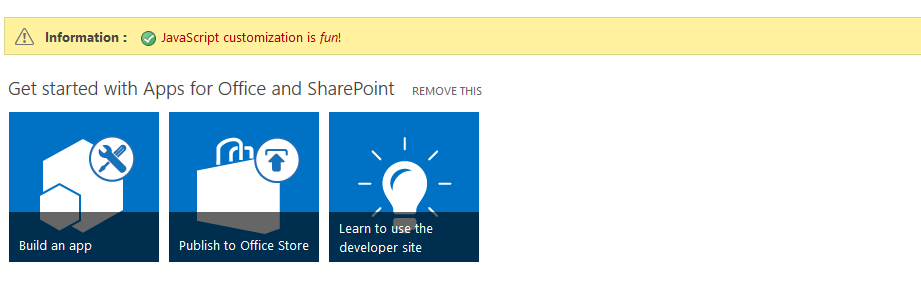
|  |  |
| --- | --- |
| Summary: | Applies to: |
| This application sample demonstrates how to perform JavaScript injection to update SharePoint sites. This approach is preferred over customizing the Master Page. | * Office 365 Multi-Tenant (MT) * Office 365 Dedicated (D) * SharePoint 2013 on-premises |
| Solution: | Contoso.Core.JavaScriptInjection, version 1.0 |
| Author: | Vesa Juvonen, Bert Jansen, Frank Marasco (Microsoft) |
| //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  // THIS CODE IS PROVIDED \*AS IS\* WITHOUT WARRANTY OF  // ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING ANY  // IMPLIED WARRANTIES OF FITNESS FOR A PARTICULAR  // PURPOSE, MERCHANTABILITY, OR NON-INFRINGEMENT.  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* | |

## Version log

Change log between different releases.

|  |  |
| --- | --- |
| Version | Notes |
| 1.0 | Initial |

# General comments

This sample shows how one can use JavaScript to perform basic updates to a SharePoint site. In this sample we will modify the Status Bar for all pages. 

# SCENARIo: JavaSCRIPT INJECTION

SharePoint team sites by default make use of the Minimal Download Strategy (MDS) technique to improve performance. If we want to load custom JavaScript files we have to take this in account by loading the scripts via the below pattern:

// Register script for MDS if possible

RegisterModuleInit("scenario1.js", RemoteManager\_Inject); //MDS registration

RemoteManager\_Inject(); //non MDS run

if (typeof (Sys) != "undefined" && Boolean(Sys) && Boolean(Sys.Application)) {

Sys.Application.notifyScriptLoaded();

}

if (typeof (NotifyScriptLoadedAndExecuteWaitingJobs) == "function") {

NotifyScriptLoadedAndExecuteWaitingJobs("scenario1.js");

}

When the page that contains your script is loaded either the MDS engine (when MDS is enabled) launches your main function (RemoteManager\_Inject) or your function is launched directly for non MDS invocations. The function that’s called is your entry point to load other scripts and to perform the required customizations. Loading other scripts often is needed: the sample shows how you can load the popular jQuery library. When loading other scripts it’s important that the script parts that depend on the loaded script are only executed after the other script was loaded and this is guaranteed via the below construct:

function RemoteManager\_Inject() {

var jQuery = "https://ajax.aspnetcdn.com/ajax/jQuery/jquery-2.0.2.min.js";

// load jQuery

loadScript(jQuery, function () {

var message = "<img src='/\_Layouts/Images/STS\_ListItem\_43216.gif' align='absmiddle'> <font color='#AA0000'>JavaScript customization is <i>fun</i>!</font>"

SetStatusBar(message);

// Customize the viewlsts.aspx page

if (IsOnPage("viewlsts.aspx")) {

//hide the subsites link on the viewlsts.aspx page

$("#createnewsite").parent().hide();

}

});

}

function SetStatusBar(message) {

var strStatusID = SP.UI.Status.addStatus("Information : ", message, true);

SP.UI.Status.setStatusPriColor(strStatusID, "yellow");

}

# Dependencies

* Microsoft.SharePoint.Client.dll
* Microsoft.SharePoint.Client.Runtime.dll