# How to intercept JavaScript alerts in WebView and display the alert messages in Windows Store apps.

## Introduction

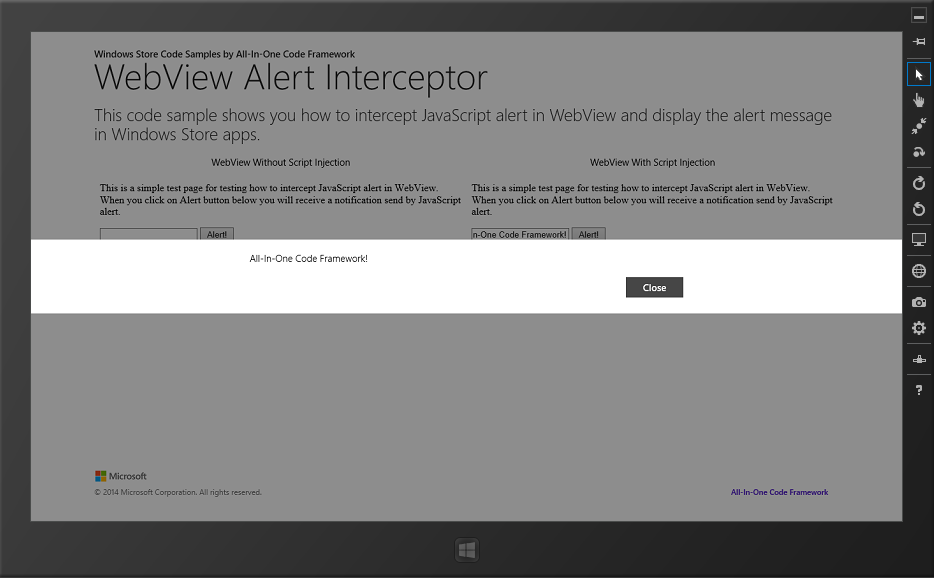
The Windows 8 & 8.1 WebView control will not show JavaScript Alerts that are raised by the webpage inside the WebView. This presents a challenge to application developers when they do not control the content of the website. However, this can be remedied by the use of JavaScript invoked into the WebView. This code sample will show you how to intercept JavaScript alerts in WebView and display the alert messages in Windows Store apps.

**Please note: The samples were written based on the following video:**

<http://channel9.msdn.com/Series/Windows-Store-Developer-Solutions/WebView-Magic-Tricks-Series-Part-3-Alert-Interceptor>

## Running the Sample

You can build and run the sample in Visual Studio 2013. After being launched, the app looks like this:



## Using the Code

The Windows 8 & 8.1 WebView control will not show Javascript alerts that are raised by the webpage inside the WebView.

scriptNotify\_example.html mimics a scenario in which the Javascript Alert() is called.

|  |
| --- |
| -Code block start-  --HTML code snippet start--  <!DOCTYPE html>  <html>  <head>  <title>Example HTML document</title>  </head>  <body onload="onload();">  <p>This is a simple test page for testing how to intercept JavaScript alerts in WebView. When you click on Alert button below, you will receive a notification sent by JavaScript alert.</p>  <input type="text" id="AlertMessageTextBox"/>  <input type="button" onclick="Alert();" value="Alert!"/>  <script type="text/javascript">  function Alert() {  window.alert(document.getElementById('AlertMessageTextBox').value);  }  </script>  </body>  </html>  --HTML code snippet end--  Insert other Programming Language Code Snippet here  -Code block end- |

Normally, the WebView control will not show JavaScript Alerts. To remedy this limitation, we can inject JavaScript code by calling **WebView.InvokeScriptAsync** function. A hosted HTML page can fire the **ScriptNotify** event in your Windows Store app when the page calls **window.external.notify** and passes a string parameter.

|  |
| --- |
| -Code block start-  --C# code snippet start--  private async void WebViewWithJSInjection\_NavigationCompleted(WebView sender, WebViewNavigationCompletedEventArgs args)  {  string result = await this.WebViewWithJSInjection.InvokeScriptAsync("eval", new string[] { "window.alert = function (AlertMessage) {window.external.notify(AlertMessage)}" });  }  --C# code snippet end--  --VB code snippet start--  Private Async Sub WebViewWithJSInjection\_NavigationCompleted(sender As WebView, args As WebViewNavigationCompletedEventArgs)  Dim result As String = Await Me.WebViewWithJSInjection.InvokeScriptAsync("eval", New String() {"window.alert = function (AlertMessage) {window.external.notify(AlertMessage)}"})  End Sub  --VB code snippet end--  --C++ code snippet start--  void MainPage::WebViewWithJSInjection\_NavigationCompleted(Windows::UI::Xaml::Controls::WebView^ sender, Windows::UI::Xaml::Controls::WebViewNavigationCompletedEventArgs^ args)  {  Array<String^>^ argsArray = { "window.alert = function (AlertMessage) {window.external.notify(AlertMessage)}" };  Platform::Collections::Vector<String^>^ arguments = ref new Platform::Collections::Vector<String^>(argsArray);  this->WebViewWithJSInjection->InvokeScriptAsync("eval", arguments);  }  --C++ code snippet end--  -Code block end- |

Then in **ScriptNotify** event handler, we can receive the alert message and show it like below.

|  |
| --- |
| -Code block start-  --C# code snippet start--  private async void WebViewWithJSInjection\_ScriptNotify(object sender, NotifyEventArgs e)  {  Windows.UI.Popups.MessageDialog dialog = new Windows.UI.Popups.MessageDialog(e.Value);  await dialog.ShowAsync();  }  --C# code snippet end--  --VB code snippet start--  Private Async Sub WebViewWithJSInjection\_ScriptNotify(sender As Object, e As NotifyEventArgs)  Dim dialog As New Windows.UI.Popups.MessageDialog(e.Value)  Await dialog.ShowAsync()  End Sub  --VB code snippet end--  --C++ code snippet start--  void MainPage::WebViewWithJSInjection\_ScriptNotify(Platform::Object^ sender, Windows::UI::Xaml::Controls::NotifyEventArgs^ e)  {  Windows::UI::Popups::MessageDialog^ dialog = ref new Windows::UI::Popups::MessageDialog(e->Value);  dialog->ShowAsync();  }  --C++ code snippet end--  -Code block end- |

## More Information

WebView Magic Tricks Series Part 3: Alert Interceptor

<http://channel9.msdn.com/Series/Windows-Store-Developer-Solutions/WebView-Magic-Tricks-Series-Part-3-Alert-Interceptor>

WebView class

<http://msdn.microsoft.com/library/windows/apps/windows.ui.xaml.controls.webview.aspx>

WebView.InvokeScriptAsync method

<http://msdn.microsoft.com/en-us/library/windows/apps/windows.ui.xaml.controls.webview.invokescriptasync.aspx>

WebView.ScriptNotify event

<http://msdn.microsoft.com/en-us/library/windows/apps/windows.ui.xaml.controls.webview.scriptnotify.aspx>