

# How can I prevent a Windows Runtime WebView from loading any content beyond the initial request?

---

 [devblogs.microsoft.com/oldnewthing/20210218-00](https://devblogs.microsoft.com/oldnewthing/20210218-00)

February 18, 2021



Raymond Chen

A customer wanted to navigate a XAML WebView control to a site and load only the HTML. No external script. No images. No CSS. Just the raw HTML returned from a single web request.

You can do this by handling the `WebResourceRequested` event. Let's take [the WebView sample](#) and make these changes.

```

public Scenario1_NavToUrl()
{
    this.InitializeComponent();
    WebViewControl.WebResourceRequested += OnResourceRequested;
}

Uri allowedUri = null;

void OnResourceRequested(WebView sender,
    WebViewWebResourceRequestedEventArgs e)
{
    if (e.Request.RequestUri != allowedUri)
    {
        e.Response = new Windows.Web.Http.HttpResponseMessage(
            Windows.Web.Http.HttpStatusCode.NotFound);
    }
}

private void NavigateWebview(string url)
{
    try
    {
        Uri targetUri = new Uri(url);
        allowedUri = targetUri; // remember where we're going
        WebViewControl.Navigate(targetUri);
    }
    catch (UriFormatException ex)
    {
        // Bad address
        AppendLog($"Address is invalid, try again. Error: {ex.Message}.");
    }
}

```

When we navigate the WebView control, we remember the target URI in the `allowedUri` member variable. When the WebView is about to download some content, it raises the `WebResourceRequested` event to let the app know what is about to happen and give the opportunity to handle the request explicitly.

In our case, we see whether the URI matches the `allowedUri`. If not, then we create a custom response that consists of error 404 (*Not found*). Maybe that's not the best error code, but I'll let you pick the error you want.

Run the sample program and enter a URL like `https://visualstudio.microsoft.com/`. The main HTML content loads, but all the other content is blocked.

On the other hand, if you use a URL like `http://www.microsoft.com`, then nothing loads at all, because `http://www.microsoft.com` is a redirect to `https://www.microsoft.com`, and since that doesn't match the `allowedUri`, we block it.

Maybe that's what you want. But if you want to allow redirects, you'll have to follow the redirections and allow them, too. We'll do that next time.

Raymond Chen

**Follow**

