



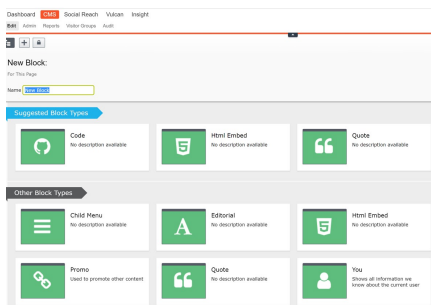
ALLAN THRAEN

5 years ago



Tips and Tricks Optimizely (Episerver) CMS Addon Development .NET Development

ADMIN MODE PLUGIN TO MANAGE CONTENT TYPE SUGGESTIONS



If you have a site with a lot of different content types, it can be a good idea to help Episervers Automatic Content Type suggestion feature along. Here is a basic Admin mode tool - in good old webforms (yes, I washed my hands after I made it) that will let administrators / and super-editors configure exactly which content types to suggest when.

As often before, I was inspired by EMVP Alf Nilsson, who wrote a [great post](#) with examples on how to extend the suggestions as a developer. But it occurred to me, that many editors I know would love the opportunity to pimp their edit-mode by customizing the content type suggestions!

If I look at this very blog, for example, I have [in the past](#) described how it's structured - but that structure also means that by default Episerver won't suggest a "blog post" when I create a new item on the Blog Root. Until now, that is. Or - when I'm working on a new blog post and start to add blocks I usually prefer Code block, HTML Embed blocks or a Quote block - which I now also can have suggested right at the top, as you can see in the top image (also notice the nice fontawesome icons thanks to GETA).

For the admin mode plugin the easiest approach was to do an oldfashioned ASPX page - so that's what I did. This is what it looks like:

Suggested Types Manager

New Custom Type Suggestion

Parent Content Type:

Select specific page:

Only suggest this for me: ☐

Is this a 'For This Content' folder?: ☐

Content Type to Suggest:

Add

Parent Content Type	Parent	Only for user	Content Folder	Content Type to Suggest	
[] BlogPostPage	Blog	False		[] BlogPostPage	
[] BlogPostPage		True		[] CodeBlock	
[] BlogPostPage		True		[] CodeEmbedBlock	
[] BlogPostPage		True		[] QuoteBlock	

I embedded the Gist below.

If you think this could be useful in the nuget feed, be sure to drop a comment below and let me know!

```
1 using EPIServer;
2 using EPIServer.Cms.Shell.UI.Rest;
3 using EPIServer.Core;
4 using EPIServer.DataAbstraction;
5 using EPIServer.Personalization;
6 using EPIServer.ServiceLocation;
7 using System;
8 using System.Collections.Generic;
9 using System.Linq;
10 using System.Web;
11
12 namespace CodeArt.Addons.ContentTypeSuggestions
13 {
14     [ServiceConfiguration(typeof(IContentTypeAdvisor))]
15     public class CustomContentRecommendations : IContentTypeAdvisor
16     {
17         public IContentRepository repo { get; set; }
18         public IContentTypeRepository typerepo { get; set; }
19
20         public CustomContentRecommendations(IContentRepository repo, IContentTypeRepository trepo)
21         {
22             this.repo = repo;
23             typerepo = trepo;
24         }
25
26         //Add custom recommendations where needed.
27         public IEnumerable<int> GetSuggestions(IContent parent, bool contentFolder, IEnumerable<string> requestedTypes)
28         {
29             foreach (var s in TypeSuggestion.List(parent.ContentTypeId, parent.ContentLink.ToReferenceWithoutVersion(), EPIServerProfile.Current.UserName))
30             {
31                 if (contentFolder == s.ContentFolder)
32                 {
33                     yield return s.TypeToSuggest;
34                 }
35             }
36         }
37     }
38 }
39
40
41 }
```

```
1 <%@ Page Language="C#" CodeBehind="SuggestedTypesAdmin.aspx.cs" AutoEventWireup="False" EnableViewState="true" Inherits="CodeArt.Addons.ContentType
2 <%@ Register TagPrefix="EPI" Server="EPI" Namespace="EPI.Server.Web.PropertyControls" Assembly="EPI.Server.Cms.AspNet" %>
3 <%@ Register TagPrefix="EPI" Server="EPI" Namespace="EPI.Server.UI.Web.Controls" Assembly="EPI.Server.UI" %>
4 <asp:Content ContentPlaceHolderID="MainRegion" runat="server">
5 <div class="epi-formArea">
6
7 <fieldset>
8 <legend>
9 <asp:Literal Text="New Custom Type Suggestion" runat="server"/>
10 </legend>
11 <div class="epi-size0">
12 <div>
13 <asp:Label runat="server" AssociatedControlID="pageTypeList" Text="Parent Content Type" />
14 <asp:DropDownList ID="pageTypeList" runat="server" />
15 </div>
16 <div>
17 <asp:label runat="server" Text="Select specific page" AssociatedControlID="parentRef" />
18 <EPI:ServerInputPageReference CssClass="epiinlineinputcontrol" ID="parentRef" runat="server" DisableCurrentPageOption="true"/><br />
19 </div>
20 <div>
21 <asp:Label runat="server" AssociatedControlID="justForMe" Text="Only suggest this for me" />
22 <asp:CheckBox runat="server" ID="justForMe" />
23 </div>
24 <div>
25 <asp:Label runat="server" AssociatedControlID="contentFolder" Text="Is this a 'For This Content' folder?" />
26 <asp:CheckBox runat="server" ID="contentFolder" />
27 </div>
28 <div>
29 <asp:Label runat="server" AssociatedControlID="typesToSuggest" Text="Content Type to Suggest" />
30 <asp:DropDownList runat="server" ID="typesToSuggest" />
31 </div>
32 </div>
33 </fieldset>
34 <div class="epi-toolbuttonrow">
35 <EPI:ServerUI:ToolButton runat="server" Text="Add" ToolTip="" OnClick="AddBtn_Click" Id="AddBtn" />
36 </div>
37
38 </div>
39 <div class="epi-formArea" ID="TypeListDiv" runat="server">
40 <asp:GridView
41 ID="TypeList"
42 Runat="server"
43 AutoGeneratesColumns="false">
44 <Columns>
45 <asp:TemplateField HeaderText="Parent Content Type">
46 <itemTemplate>
47 <%#: GetContentTypeName((int)DataBinder.Eval(Container.DataItem, "ParentType"))%>
48 </itemTemplate>
49 </asp:TemplateField>
50 <asp:TemplateField HeaderText="Parent">
51 <itemTemplate>
52 <%#: GetContentName((EPI:Server.Core.ContentReference)DataBinder.Eval(Container.DataItem, "ParentReference"))%>
53 </itemTemplate>
54 </asp:TemplateField>
55 <asp:TemplateField HeaderText="Only for user">
56 <itemTemplate>
57 <%# DataBinder.Eval(Container.DataItem, "User")%>
58 </itemTemplate>
59 </asp:TemplateField>
60 <asp:TemplateField HeaderText="Content Folder">
61 <itemTemplate>
62 <%# DataBinder.Eval(Container.DataItem, "ContentFolder")%>
63 </itemTemplate>
64 </asp:TemplateField>
65 <asp:TemplateField HeaderText="Content Type to Suggest">
66 <itemTemplate>
67 <%#: GetContentTypeName((int)DataBinder.Eval(Container.DataItem, "TypeToSuggest"))%>
68 </itemTemplate>
69 </asp:TemplateField>
70 <asp:TemplateField>
71 <itemTemplate>
72 <EPI:ServerUI:ToolButton EnableClientConfirm="true" ID="DeleteToolButton" CommandName="Delete" SkinID="Delete"
73 ToolTip="%%$ Resources: EPI:Server, button.delete %%" Enabled="true" OnCommand="DeleteToolButton_Command" CommandArgument="<
74 </itemTemplate>
75 </asp:TemplateField>
76 </Columns>
77 </asp:GridView>
78 </div>
79
80 </asp:Content>
```

SuggestedTypesAdmin.aspx hosted with ❤ by GitHub

[view raw](#)

```
1 using System;
2 using System.Collections.Generic;
3 using System.Web.Security;
4 using System.Web.UI.WebControls;
5 using EPI:Server.Personalization;
6 using EPI:Server.Plugin;
7 using EPI:Server.Security;
8 using EPI:Server.Util.Plugins;
9 using System.Web.UI;
10 using EPI:Server.UI;
11 using EPI:Server.ServiceLocation;
12 using EPI:Server.DataAbstraction;
13 using System.Linq;
14 using EPI:Server.Core;
15 using EPI:Server;
16 using EPI:Server.Data;
17
18 namespace CodeArt.Addons.ContentTypeSuggestions
19 {
20 [GuiPlugin(DisplayName = "Suggested Types Manager", Description = "", Area = PluginArea.AdminMenu, UrlFromUI = "/CodeArt.Addons.ContentTypeSuggestions
21 public partial class SuggestedTypesAdmin : EPI:Server.Shell.WebForms.WebFormsBase
22 {
23
24     protected override void OnLoad(EventArgs e)
25     {
26         base.OnLoad(e);
27         this.EnableViewState = true;
28         this.Title = "Suggested Types Manager";
29         this.SystemMessageContainer.Heading = "Suggested Types Manager";
30         if (IsPostBack)
31         {
32             var trepo = ServiceLocator.Current.GetInstance<IContentTypeRepository>();
33             this.pageTypeList.DataSource = new List<Item> { new Listitem("All types"), "-1" }
34             .Union(trepo.List(OrderBy(c => c.FullName).Select(pt => new Listitem(pt.FullName, pt.ID.ToString())));
35             this.pageTypeList.DataTextField = "Value";
36             this.pageTypeList.DataTextFields = "Text";
37             this.TypesToSuggest.DataSource = trepo.List(OrderBy(c => c.FullName).Select(pt => new Listitem(pt.FullName, pt.ID.ToString()));
38             this.TypesToSuggest.DataValueField = "Value";
39             this.TypesToSuggest.DataTextFields = "Text";
```

```
40 //Show existing list
41 this.TypeListDataSource = TypeSuggestion.ListForUser(EpiServerProfile.Current.UserName);
42 this.DataBind();
43 }
44 }
45
46
47 public string GetContentTypeName(int ID)
48 {
49     if (ID == -1) return "";
50     var trepo = ServiceLocator.Current.GetInstance<IContentTypeRepository>();
51     return trepo.Load(ID).FullName;
52 }
53
54 public string GetContentName(ContentReference cref)
55 {
56     if (cref == ContentReference.EmptyReference) return "";
57     var repo = ServiceLocator.Current.GetInstance<IContentTypeRepository>();
58     return repo.Get<IContent>(cref).Name;
59 }
60
61 public void AddBtn_Click(object sender, EventArgs e)
62 {
63     //Validate and Add to DDS
64     TypeSuggestion ts = new TypeSuggestion();
65     ts.ParentType = int.Parse(pageTypeList.SelectedValue);
66     ts.ParentReference = this.parentRef.PagedList ?? ContentReference.EmptyReference;
67     ts.ContentFolder = this.contentFolder.Checked;
68     ts.User = (this.justForMe.Checked) ? EpiServerProfile.Current.UserName : null;
69     ts.TypeToSuggest = int.Parse(this.TypesToSuggest.SelectedValue);
70     TypeSuggestion.Save(ts);
71     Response.Redirect(Request.Url.ToString());
72 }
73
74 protected void DeleteToolButton_Command(object sender, CommandEventArgs e)
75 {
76     TypeSuggestion.Delete(Identity.Parse((string) e.CommandArgument));
77     Response.Redirect(Request.Url.ToString());
78 }
79 }
80 }
```

SuggestedTypesAdmin.aspx.cs hosted with ❤ by GitHub [view raw](#)

```
1 using EpiServer.Core;
2 using EpiServer.Data;
3 using EpiServer.Data.Dynamic;
4 using System;
5 using System.Collections.Generic;
6 using System.Linq;
7 using System.Web;
8
9 namespace CodeArt.Addons.ContentTypeSuggestions
10 {
11     [EpiServerDataStore(AutomaticallyCreateStore = true, AutomaticallyRemapStore = true)]
12     public class TypeSuggestion : IDynamicData
13     {
14         public Identity Id { get; set; }
15
16         public int ParentType { get; set; }
17
18         public ContentReference ParentReference { get; set; }
19
20         public int TypeToSuggest { get; set; }
21
22         public string User { get; set; }
23
24         public bool ContentFolder { get; set; }
25
26         public static IEnumerable<TypeSuggestion> ListForUser(string user)
27         {
28             var store = DynamicDataStoreFactory.Instance.CreateStore<TypeSuggestion>();
29             return store.Items<TypeSuggestion>().Where(ts => ts.User == user || ts.User == null);
30         }
31
32         public static void Delete(Identity ID)
33         {
34             var store = DynamicDataStoreFactory.Instance.CreateStore<TypeSuggestion>();
35             store.Delete(ID);
36         }
37
38         public static IEnumerable<TypeSuggestion> List(int TypedId, ContentReference Parent, string User)
39         {
40             var store = DynamicDataStoreFactory.Instance.CreateStore<TypeSuggestion>();
41             return store.Items<TypeSuggestion>().Where(ts => (ts.User == User || ts.User == null)
42                 && (ts.ParentType == TypedId || ts.ParentType == -1)
43                 && (ts.ParentReference == Parent.ToReferenceWithoutVersion() || ts.ParentReference == ContentReference.EmptyReference));
44         }
45
46         public static void Save(TypeSuggestion suggestion)
47         {
48             var store = DynamicDataStoreFactory.Instance.CreateStore<TypeSuggestion>();
49             store.Save(suggestion);
50         }
51     }
52 }
```

TypeSuggestion.cs hosted with ❤ by GitHub [view raw](#)

Tips and Tricks | [Optimizely \(EpiServer\)](#) | [CMS](#) | [Addon Development](#) | [.NET Development](#)

RECENT POSTS

Alf's post