



ALLAN THRAEN

4 years ago

[Tips and Tricks](#) [Tech Talk](#) [CMS](#) [Optimizely \(EpiServer\)](#)

## ASCEND 2019: CODE MANIA



**Yesterday, I had the honor and pleasure of giving the traditional Code Mania demo at EpiServer Ascend 2019 in Miami together with Fredrik Haglund. After popular demand, here is a blog post about some of the components we showed.**

In the middle of the Miami heat, Fredrik and I was asked to round off the Developer Track with the traditional Code Mania session. Code Mania sessions are notorious for not having any slides and just being live demos with a fun twist - and this year was no different. But after having been approached by quite a few attendees afterwards, I was persuaded to share this little blog post with links to the relevant components we showed - where possible.




The theme this year was about how we as developers can show editors our love - and perhaps make their lives a bit easier. Cause happy editors produce better content. And good content is what it's all about.

Here's the agenda we started to show (I'm not going to share the code for that, as I'm certain you, dear reader, is perfectly capable of provoking similar exceptions).

```
[NullReferenceException: Object reference not set to an instance of an object]
CodeMania.AgendaPageController.Introduction() in C:\Git\CodeMania.AgendaPageController.Gamification() in C:\Git\CodeMania.AgendaPageController.ContentCreation() in C:\Git\CodeMania.AgendaPageController.PublishLock() in C:\Git\CodeMania.AgendaPageController.EditingContent() in C:\Git\CodeMania.AgendaPageController.PropertyInheritance() in C:\Git\CodeMania.AgendaPageController.DeveloperTools() in C:\Git\CodeMania.AgendaPageController.WrapUpAndThankYou() in
```

### Gamifying the editorial experience: EpiServers Cat

First off, we thought - why not make the editor life a bit more exciting. What if we could gamify the experience of being an editor in EpiServer? And what if we could do it using something as cute as kittens? Now, you may have heard about Schrödinger's cat - but what if we let loose a kitten in EpiServer - and our beloved editors would have to find it? That sounded like fun.



[Start](#) [Alloy Plan!!](#) [Alloy Track](#) [Alloy Meet](#) [About us](#) [Agenda](#)


[Start](#) / [About us](#) / [Management](#)

News & Events

Management


Contact us

Become a reseller



## Management

Our experienced and dedicated managers are coming to you to ensure the effectiveness of project teams everywhere. Meet them here.




**Robert Carlsson**

Robert is Alloy's CEO and visionary leader. Robert has led technology development for very complex projects for over 20 years. Previously, he was a director at the Marshall Space Flight Center.

E-mail: [robert.carlsson@alloytech.biz](mailto:robert.carlsson@alloytech.biz)

Phone: +46 8 123 456



**Michelle**

Michelle brings operational man... been the CFO for...

E-mail: [michelle@alloytech.biz](mailto:michelle@alloytech.biz)

This will melt hearts, I tell you!

Now, the code for this was based on a sample addon for a lab about EpiServers Marketplace. While it's a bit modified in the CodeMania version, you can check out the lab [github](#) here.

The CodeArt specific version (minus the kitten) is also in this [gist](#):

```
1 using CodeMania.Site.Business;
2 using EpiServer;
3 using EpiServer.Core;
4 using EpiServer.Framework.Cache;
5 using EpiServer.Framework.Web.Resources;
6 using EpiServer.Logging;
7 using EpiServer.Security;
```

```

8 using EPIServer.ServiceLocation;
9 using EPIServer.Web.Routing;
10 using System;
11 using System.Collections.Generic;
12 using System.Linq;
13 using System.Web;
14
15 namespace CodeMania.Site.EpiKitty
16 {
17     [ClientResourceRegistrar]
18     public class CatFun : IClientResourceRegistrar
19     {
20         private static readonly ILogger logger = LogManager.GetLogger();
21
22         private IContentRouteHelper _routeHelper;
23         private IContentRepository _repo;
24         private ISynchronizedObjectInstanceCache _cache;
25
26         public ContentReference HidingPlace
27         {
28             get
29             {
30                 var cref = _cache.Get<ContentReference>("Kitty", ReadStrategy.Immediate);
31                 if (cref == null)
32                 {
33                     var lst = _repo.GetDescendents(ContentReference.StartPage)
34                         .Take(200).Where(cr => !cr.CompareToIgnoreWorkID(ContentReference.StartPage))
35                         .Select(cr => _repo.Get<PageData>(cr)).FilterForDisplay(true, true).ToList();
36                     Random R = new Random();
37                     var page = lst[R.Next(0, lst.Count)];
38                     cref = page.ContentLink;
39                     _cache.Insert("Kitty", cref, new CacheEvictionPolicy(new TimeSpan(2, 0, 0), CacheTimeoutType.Absolute));
40                     logger.Error($"Kitty is hiding on page '{page.Name}' with ID '{page.ContentLink}");
41                 }
42                 return cref;
43             }
44         }
45
46         public CatFun(IContentRouteHelper crh, IContentRepository repo, ISynchronizedObjectInstanceCache cache)
47         {
48             _routeHelper = crh;
49             _repo = repo;
50             _cache = cache;
51         }
52
53         public void RegisterResources(IRequiredClientResourceList requiredResources)
54         {
55             if (PrincipalInfo.HasEditAccess)
56             {
57                 var content = _routeHelper.Content;
58                 if (content != null && content.ContentLink.CompareToIgnoreWorkID(HidingPlace))
59                 {
60                     requiredResources.RequireScript("/EpiKitty/catfun.js");
61                 }
62             }
63         }
64     }
65 }

```

CatFun.cs hosted with ♥ by GitHub

[view raw](#)

```

1 function getRandomPosition(element) {
2     var vw = Math.max(document.documentElement.clientWidth, window.innerWidth || 0);
3     var vh = Math.max(document.documentElement.clientHeight, window.innerHeight || 0);
4     var maxY = vh - (element.clientHeight / 3);
5     var maxX = vw - element.clientWidth;
6     var randomX = Math.floor(Math.random() * maxX);
7     var randomY = Math.floor(Math.random() * maxY);
8     return [randomX, randomY];
9 }
10 window.onload = function() {
11     var img = document.createElement('img');
12     img.setAttribute("style", "position: absolute;");
13     img.setAttribute("src", "/EpiKitty/kitten.png");
14     img.setAttribute("width", 267);
15     img.setAttribute("height", 300);
16     document.body.appendChild(img);
17     img.onclick = function () { alert("Yeah! You found me! Well done. Come back later and see if you can find my new hiding place.");};
18     var xy = getRandomPosition(img);
19     img.style.left = xy[0] + 'px';
20     img.style.top = xy[1] + 'px';
21 }

```

CatFun.js hosted with ♥ by GitHub

[view raw](#)

## Property Lottery

The only thing editors love more than kittens is probably finding things to complain about. So we thought, why not indulge them and provide them a little bug, that is not straightforward to identify - it will kind of help you separate the sheep from the goats.

By adding an attribute to any property on a model, we created a piece of code that will hide it from edit-mode every 10 seconds.

Here it is - use it with care:

```

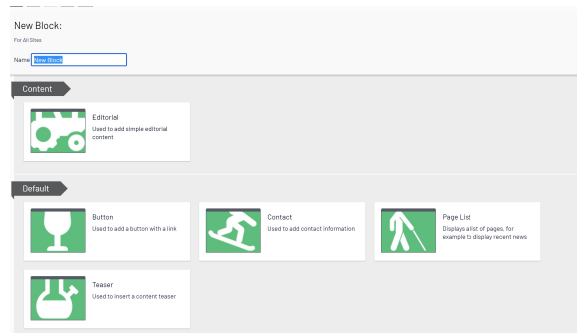
1 using EPIServer.Oms.Shell.Extensions;
2 using EPIServer.Shell.ObjectEditing;
3 using System;
4 using System.Collections.Generic;
5 using System.Linq;
6 using System.Web;
7 using System.Web.Mvc;
8
9 namespace CodeMania.Site.Fun
10 {
11     [AttributeUsage(AttributeTargets.Property, AllowMultiple = false)]
12     public class PropertyLotteryAttribute : Attribute, IMetadataAware
13     {
14         public void OnMetadataCreated(ModelMetadata metadata)
15         {
16             ExtendedMetadata extendedMetadata = metadata as ExtendedMetadata;
17             if (extendedMetadata == null) return;
18             extendedMetadata.ShowForEdit = ((DateTime.Now.Second / 10) % 2 == 0);
19         }
20     }
21 }

```

PropertyLottery.cs hosted with ♥ by GitHub

[view raw](#)

When creating new content, we showed how Geta's Font Awesome Thumbnail add-on makes it really easy to have nice icons to help your editors pick the right content. It's available in the nuget feed and here on [github](#).



## Content Type Suggestions

This is from a [blog post](#) of mine. But it's also in the [Gist](#) here.

```
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
CustomContentTypeSuggestion.cs hosted with ❤ by GitHub
```

```
1 <%@ Page Language="c#" Codebehind="SuggestedTypesAdmin.aspx.cs" AutoEventWireup="false" EnableViewState="true" Inherits="CodeArt.Addons.ContentType
2 <%@ Register TagPrefix="EPiServer" Namespace="EPiServer.Web.PropertyControls" Assembly="EPiServer.Cms.AspNet" %>
3 <%@ Register TagPrefix="EPiServerUI" Namespace="EPiServer.UI.WebControls" Assembly="EPiServer.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 <EPiServer.InputPageReference CssClass="epiinlineinputcontrol" ID="parentRef" runat="server" DisableCurrentPageOptions="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 <EPiServerUI: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 AutoGenerateColumns="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 <%# GetContentTypeName((EpiServer.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 <EpiServer.UI.ToolButton EnableClientConfirm="true" ID="DeleteToolButton" CommandName="Delete" SkinID="Delete"
73 ToolTip="<%%$ Resources: EpiServer, 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 EpiServer.Personalization;
6 using EpiServer.Plugin;
7 using EpiServer.Security;
8 using EpiServer.Util.Plugins;
9 using System.Web.UI;
10 using EpiServer.UI;
11 using EpiServer.ServiceLocation;
12 using EpiServer.DataAbstraction;
13 using System.Linq;
14 using EpiServer.Core;
15 using EpiServer;
16 using EpiServer.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 : EpiServer.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 List<Item>("All types"), "-1") }
34                 .Union(trepo.List(0, OrderBy(c => c.FullName).Select(pt => new List<Item>(pt.FullName, pt.ID.ToString())));
35                 this.pageTypeList.DataValueField = "Value";
36                 this.pageTypeList.DataTextField = "Text";
37                 this.TypesToSuggest.DataSource = trepo.List(0, OrderBy(c => c.FullName).Select(pt => new List<Item>(pt.FullName, pt.ID.ToString())));
38                 this.TypesToSuggest.DataValueField = "Value";
39                 this.TypesToSuggest.DataTextField = "Text";
40                 //Show existing list
41                 this.TypeList.DataSource = TypeSuggestion.ListForUser(EpiServer.Profile.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<IContentRepository>();
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.PageLink ?? ContentReference.EmptyReference;
67             ts.ContentFolder = this.contentFolder.Checked;
68             ts.User = (this.justForMe.Checked) ? EpiServer.Profile.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;
```

```

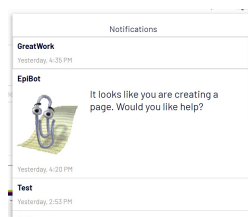
3 using EPiServer.Data.Dynamic;
4 using System;
5 using System.Collections.Generic;
6 using System.Linq;
7 using System.Web;
8
9 namespace CodeArt.Addons.ContentTypeSuggestions
10 {
11     [EPiServer.DataStore(AutomaticallyCreateStore = true, AutomaticallyRemapStore = true)]
12     public class TypeSuggestion : 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 typeId, ContentReference parent, string user)
39         {
40             var store = DynamicDataStoreFactory.Instance.CreateStore<TypeSuggestion>();
41             return store.Items<TypeSuggestion>().Where(ts => (ts.ParentType == typeId || ts.ParentType == -1)
42                 && (ts.ParentReference == parent.ToReferenceWithoutVersion() || ts.ParentReference == ContentReference.EmptyReference));
43         }
44
45         public static void Save(TypeSuggestion suggestion)
46         {
47             var store = DynamicDataStoreFactory.Instance.CreateStore<TypeSuggestion>();
48             store.Save(suggestion);
49         }
50     }

```

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

## Clippy for Episerver

If anybody knows how to make a sweet UI for editors it must be Microsoft. So we invited our old office friend Clippy back - into Episerver notifications.



The code for this is not available at the moment.

## Publish Lock

Unfortunately not all editors have strictly managed access rights - and if you are creating content in Episerver and have publish access, you are in the risk of accidentally publishing it - just one wrong click (well, ok 2) and you are in big trouble. So - inspired by the little plastic cap that's on top of the "Nuclear Launch Button" we added a feature to lock content for publishing.



The code is not currently available.

## Un-moveable page

Some times you also want to protect a page from accidentally being moved around the tree. We showed this achieved, by adding the IProtectedContent interface to the page model declaration - and it was handled by an event handler on the MovingContent event, that would cancel the move, unless you had the special permission for it. Currently the code is not available.

## Show Descriptions

Sometimes, it would be helpful to show the description of each property to the editors right away, instead of providing it as a tooltip. To accomplish that, we used the approach described in Alf's blog post here: <https://talk.alfnjlsson.se/2014/12/18/display-help-text-in-on-page-editing/>

Page description  
Used as the meta  
description and commonly  
as an Ingress

Project management has never been easier! Use Alloy Meet with Alloy Plan to get the whole team involved in the creation of project plans and see how this commitment translates into finite and achievable goals.

## Inline block editing

As it was also shown in the keynote, Episerver Labs has made some great improvements with the much desired feature inline block editing. It's available already now in the nuget feed - and we showed that as well. Here's the original blog post introducing it: <https://world.episerver.com/blogs/grzegorz-wiechec/dates/2019/7/episerver-labs---block-enhancements/>

## Property Inheritance

We showed a slightly modified version of the property inheritance approach I blogged about here:

<https://www.codeart.dk/blog/2018/9/good-ol-dynamic-properties/>

Here is the gist of the original code:

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5
6 namespace AllanTech.Web.PropertyInheritance
7 {
8     public class InheritAttribute : Attribute
9     {
10         /// <summary>
11         /// Name of Boolean property that indicates if this property should be inherited
12         /// </summary>
13         public string SwitchPropertyName { get; set; }
14
15         /// <summary>
16         /// Inherit this value if it's null
17         /// </summary>
18         public bool InheritIfNull { get; set; }
19
20         /// <summary>
21         /// Inherit this value if it's null or empty
22         /// </summary>
23         public bool InheritIfNullOrEmpty { get; set; }
24
25         /// <summary>
26         /// Name of property on parent content to inherit from. Default is same name.
27         /// </summary>
28         public string ParentPropertyToInheritFrom { get; set; }
29
30         /// <summary>
31         /// Keep searching ancestors until Root
32         /// </summary>
33         public bool SearchAllAncestors { get; set; }
34     }
35 }
36 }
```

InheritAttribute.cs hosted with ❤ by GitHub

[view raw](#)

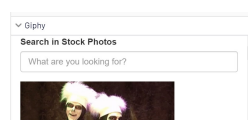
```
1 using EPiServer.Core;
2 using System;
3 using System.Collections.Generic;
4 using System.Linq;
5 using System.Web;
6 using AllanTech.Web.Helpers;
7 using EPiServer.ServiceLocation;
8 using EPiServer;
9 using EPiServer.Data.Entity;
10 using System.Reflection;
11
12 namespace AllanTech.Web.PropertyInheritance
13 {
14     public static class PropertyInheritor
15     {
16
17         public static T PopulateInheritedProperties<T>(this T Content) where T : PageData
18         {
19             var rt = (Content as IReadOnly).CreateWritableClone() as PageData;
20             var props = Content.GetPropertiesWithAttributes(typeof(InheritAttribute));
21             bool modified = false;
22             foreach (var prop in props)
23             {
24                 var attr = prop.GetCustomAttribute<InheritAttribute>(true);
25
26                 if (
27                     (String.IsNullOrEmpty(attr.SwitchPropertyName) && ((bool)Content.GetType().GetProperty(attr.SwitchPropertyName).GetValue(Content))) ||
28                     ((attr.InheritIfNull || attr.InheritIfNullOrEmpty) && (prop.GetValue(Content) == null)) ||
29                     (attr.InheritIfNullOrEmpty && ((prop.PropertyType == typeof(ContentArea) && (prop.GetValue(Content) as ContentArea).Count == 0))
30                 )
31                 {
32                     //Resolve Inherited Properties
33                     var repo = ServiceLocator.Current.GetInstance<IContentRepository>();
34                     foreach (var a in repo.GetAncestors(Content.ContentLink).Take((attr.SearchAllAncestors ? 1000 : 1))
35                     {
36                         var parentprop = (a as IContentData).Property[attr.ParentPropertyToInheritFrom ?? prop.Name];
37                         if (parentprop != null && !parentprop.IsNull)
38                         {
39                             prop.SetValue(rt, parentprop.Value);
40                             modified = true;
41                             break;
42                         }
43                     }
44                 }
45             }
46             if (modified)
47             {
48                 rt.MakeReadOnly();
49                 return rt as T;
50             }
51             return Content;
52         }
53     }
54 }
55
56 }
57 }
```

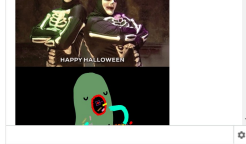
PropertyInheritor.cs hosted with ❤ by GitHub

[view raw](#)

## Giphy Images

To provide editors with an easy and efficient way to fetch fun animated gifs for their content, we showcased a very basic prototype widget integration to Giphy. The code is not available.

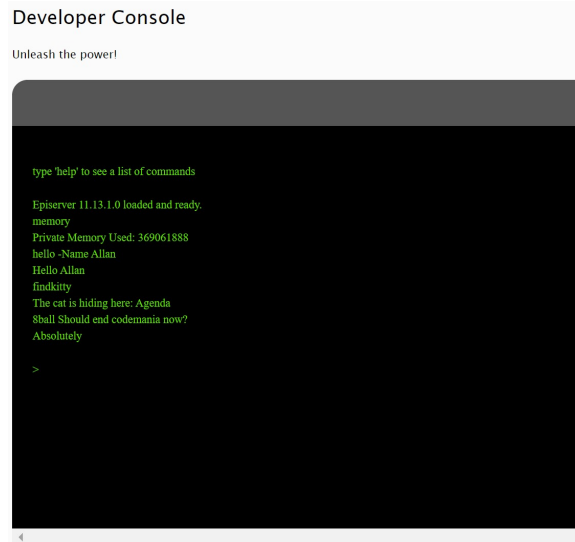




## Developer Tools & Developer Console

For the closing act, we showed the great and classic Developer Tools add-on, available in the nuget feed. However, our version was modified and is not yet publicly available. The new thing was the Developer Console - a command console that developers can use in a browser to run custom commands on Episerver - complete with a parsing engine and pipe support.

The idea is that developers instead of making scheduled jobs, only supposed to run once, can easily create commands that do what they want - and run them from the console. It's a project still in it's infancy, but it will be coming to a nuget feed near you in the future.



We hoped you enjoyed the show! If there's any of the features you particularly liked, feel free to reach out or leave a comment below!

[Tips and Tricks](#) | [Tech Talk](#) | [CMS](#) | [Optimizely \(Episerver\)](#)

## RECENT POSTS

Fredrik's blog

**CodeArt ApS**  
Teknikerbyen 5, 2830 Virum, Denmark  
**Email:** [info@codeart.dk](mailto:info@codeart.dk)  
**Phone:** +45 26 13 66 96  
**CVR:** 39680688



Copyright © 2024