



ALLAN THRAEN | 4 years ago | PDF |

Tips and Tricks C# CMS Optimizely (EpiServer)

EPISERVER FORMS: ADDING CUSTOM VALIDATORS TO EXISTING ELEMENTS

It's easy to extend Episerver forms with custom validation types. But it's a little bit harder to add the new validation types to the existing elements. In this blog post I'll add an ultra simple letter-only validation option to the existing TextboxElement.



Want to get more out of Episerver Forms?

We are experts in optimizing and extending Episerver forms. Reach out and let's talk.

[REACH OUT](#)

I got this request from a customer yesterday, and I figured more could probably benefit from this approach.

Challenge: Add a number of custom form input validation options to existing form elements.

Now, Episerver Forms is **very** extendable. And it is quite easy to add your own validation - you simply have to create something that inherits `ElementsValidatorBase` or one of it's descendents. But in order to attach it to a form element type, the typical approach is to create a new element type.

There are numerous examples of this - and Episerver offers one as well:

<https://world.episerver.com/documentation/developer-guides/forms/creating-form-element-with-validator/>

However, what if you are perfectly happy with the built-in types, and simply want to extend it with another validation type - but you can't since it's already locked down by the `[AvailableValidatorTypes]` attribute (as most of the built-in are)?

Sure, you could inherit the class and make a minimalistic new type, but soon you'll get buried in a lot of elements - and getting the editors to pick the right one might not be easy. Of course there are hacks to hide the built-in elements, but I prefer this approach.

It's pretty straight forward. The `IValidationService` is in charge of getting the validators, and in the example below, I extend it with a dictionary of additional validators that'll get added after the regular logic.

The example validator I have added is a very simple `LettersOnly` validator for the textbox element - but it could have been any kind of validator.

The only code not included in the Gist here is the language text for UI display.

```
1 using System;
2 using System.Linq;
3 using EPiServer.Forms.Core.Validation;
4 using EPiServer.Forms.Implementation.Elements;
5 using EPiServer.Framework;
6 using EPiServer.Framework.Initialization;
7 using EPiServer.ServiceLocation;
8
9 namespace AlloyExperiments.Forms
10 {
11     [InitializableModule]
12     [ModuleDependency(typeof(EPiServer.Web.InitializationModule))]
13     public class FormValidationInit : IConfigurableModule, IInitializableModule
14     {
15         public void ConfigureContainer(ServiceConfigurationContext context)
16         {
17             //Ensure we use our version of the ValidationService
18             context.StructureMap.O.Configure(c =>
19                 c.ForSingletonOf<IValidationService>().Use<ValidationServiceWithCustomValidators>());
20         }
21
22         public void Initialize(InitializationEngine context)
23         {
24             //Add initialization logic, this method is called once after CMS has been initialized
25             context.InitComplete += Context__InitComplete;
26         }
27
28         private void Context__InitComplete(object sender, EventArgs e)
29         {
30             //Add our validator as an extra validator for the TextboxElementBlock
31             var validationService = ServiceLocator.Current.GetInstance<IValidationService>();
32             (validationService as ValidationServiceWithCustomValidators).AddAdditionalValidator(typeof(TextboxElementBlock), typeof(LettersOnlyValidator));
33         }
34
35         public void Uninitialize(InitializationEngine context)
36         {
37             //Add uninitialization logic
38         }
39     }
40 }
```

39)
40)

FormValidationInit.cs hosted with ❤ by GitHubview raw

```
1 using EPiServer.Forms.Core.Models;
2 using EPiServer.Forms.Core.Validation.Internal;
3
4
5 namespace AlloyExperiments.Forms
6 {
7     public class LettersOnlyValidator : RegularExpressionValidatorBase
8     {
9         public override int ValidationOrder => 400;
10
11         public override IValidationModel Model
12         {
13             get
14             {
15                 if (this._model == null)
16                 {
17                     string str = "(^$)|(^([a-zA-Z])+)$"; // Either empty string, or only letters
18                     this._model = this.GetRegularExpressionValidationModel(str, str);
19                 }
20                 return this._model;
21             }
22         }
23     }
24 }
```

LettersOnlyValidator.cs hosted with ❤ by GitHubview raw

```
1
2 using EPiServer;
3 using EPiServer.Forms.Core.Validation;
4 using EPiServer.Forms.EditView.Internal;
5 using EPiServer.ServiceLocation;
6 using System;
7 using System.Collections.Generic;
8 using System.Linq;
9 using System.Web;
10
11 namespace AlloyExperiments.Forms
12 {
13     [ServiceConfiguration(typeof(IValidationService), Lifecycle = ServiceInstanceScope.Singleton)]
14     public class ValidationServiceWithCustomValidators : ValidationService
15     {
16         private Dictionary<Type, List<Type>> _additionalValidators;
17
18         protected Dictionary<Type, List<Type>> AdditionalValidators
19         {
20             get
21             {
22                 if (_additionalValidators == null) _additionalValidators = new Dictionary<Type, List<Type>>();
23                 return _additionalValidators;
24             }
25         }
26
27         public void AddAdditionalValidator(Type ElementType, Type ValidatorType)
28         {
29             if (AdditionalValidators.ContainsKey(ElementType)) AdditionalValidators.Add(ElementType, new List<Type>() {
30                 AdditionalValidators[ElementType].Add(ValidatorType);
31             }
32
33             public override IEnumerable<Type> GetValidatorTypesForAnElementTypeInEditView(ElementValidatable element)
34             {
35                 var rt = base.GetValidatorTypesForAnElementTypeInEditView(element);
36                 if (AdditionalValidators.ContainsKey(element.GetOriginalType0))
37                 {
38                     var lst = AdditionalValidators[element.GetOriginalType0];
39                     rt = rt.Union(lst).Distinct();
40                 }
41
42                 return rt;
43             }
44         }
45     }
```

ValidationServiceWithCustomValidators.cs hosted with ❤ by GitHubview raw

Tips and Tricks

C#

CMS

Optimizely (EpiServer)

RECENT POSTS

Episerver Forms: Adding Datasource for hidden field with Profile Store data

CodeArt ApS

Teknikerbyen 5, 2830 Virum, Denmark

Email: info@codeart.dk

Phone: +45 26 13 66 96

CVR: 39680688

in

Copyright © 2024