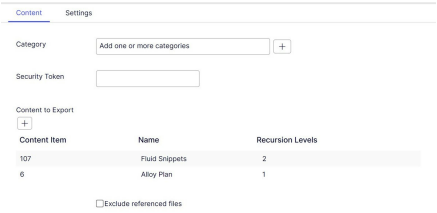


ALLAN THRAEN | 1 years ago | PDF |

Tips and TricksC#Website ImprovementsOptimizely (Episerver)CMS

CUSTOMIZING  
PROPERTY LISTS IN  
OPTIMIZEZLY CMS



**Generic property lists is a cool editorial feature that has gained a lot of popularity - in spite of still being unsupported (officially). But if you dig a bit deeper you can actually customize the editor experience even more...**

The ability to have properties that are lists of any complex object you can think of is incredibly powerful when building nice editor experiences. Historically it started as an internal functionality that was used in several projects - for example Episerver Forms, for those cases where a full blown content area with blocks is simply overkill - but a single property won't do it.

Today it's fairly well described - even in Optimizely's own documentation (<https://docs.developers.optimizely.com/content-cloud/v12.0.0-content-cloud/docs/generic-propertylist>) however, still officially unsupported.

As many of you may know, to get going with a custom property list you need a 3 things:

- The custom POCO object that describes each instance. Here you can use most of the usual attributes we normally use when describing properties to help with customize the user experience
  - A [PropertyDefinitionTypePlugin] - usually an empty class that simply inherits PropertyList<CustomObject>.
  - To add the Property to the ContentType - usually as an IList<CustomObject> and with an attribute like this: [EditorDescriptor(EditorDescriptorType = typeof(CollectionEditorDescriptor<CustomObject>))].
- So far, so good.

But what if you want to customize even more? I am working on a case where I more fields on my custom object than I want to show as columns in a grid. First attempt was to add the good, old [ScaffoldColumn] to the properties that should be hidden. And it did hide them - but only in the Edit Item view, not in the grid.

It turns out, that if you inherit the CollectionEditorDescriptor<CustomObject> and replace it with your own subclass, then you can setup a lot more useful settings in the properties. For example, it has a GridDefinition that very practically holds a list of Excluded Column names - and that did the trick.

You can do much more there, though:

- Control the commands for each item
- Manage texts for dialogs
- Turn sorting on/off
- Handle drag and drop to the list.

Tips and TricksC#Website ImprovementsOptimizely (Episerver)CMS

RECENT POSTS