

 ALLAN THRAEN | 3 years ago | PDF |

Optimizely (Episerver) C# Tips and Tricks

# SHOWING THE RIGHT BLOCK TYPES AT THE RIGHT TIME



Often, it can be handy to have very exact and dynamic control over what editors can see or do. One of my weapons of choice for this is the good, old and trusted Virtual Roles. In this scenario we have a multi-site Episerver, where some of the block types it should only be possible to use on one of the sites.

5-10 years ago I remember showing a demo of how to test your users ability to make accurate bug reports. The approach was simple: Configure a virtual role to only be true when `DateTime.Now.Seconds % 2 == 0`. Or in other words, every other second the user was in the role - and every other second they were not. All that was left to do was to make a certain property on a certain page visibility in edit mode, only to those in the role :-). The funny thing was that some editors would start to rationalize on the reasoning on why sometimes they would see it and sometimes they wouldn't. But I digress. Virtual Roles is a hidden gem in Episerver and you can do a lot of really powerful things with them. In fact, they were the original precursor to Visitor Groups and personalization. And the beauty is that anything in Episerver you can manage with permissions and roles, you can use manage with dynamic code in a virtual role.

How about creating a role that's the only one with "Publish" permissions that's only in the role for editors between 9 and 3pm to avoid those late night mistakes? Or maybe a role with access to confidential information that also requires you to be on corporate premises (IP)?

I was asked to help out a client with a case where there are multiple campaign sites running on the same Episerver instance. However, they are quite different and not all block types work well on all sites. Now, of course you can (and to a certain degree should) restrict which block types can be referenced in a content area using `[AllowedTypes]` attribute - but that doesn't prevent an editor from still creating the wrong type and spending time and effort building the wrong kinds of content. You can of course promote certain types in the "Suggested Types" - but it's still good practice to simply remove the types from the list that they can't use on the given site.

To solve that, I simply create the virtual role, seen in the Gist below. Then it was simply a matter of adding it in the web.config.

```
<episerver.framework>
<virtualRoles addClaims="true">
  <providers>
    ...
    <add name="SiteA" type="MyExperiments.VirtualRoles.SiteVirtualRole,MyExperiments" site="sitea" />
  </providers>
</virtualRoles>
...
</episerver.framework>
```

And of course restricting the appropriate block types with the `[Access]` attribute

```
[SiteContentType(GUID = "426CF12F-1F01-4EA0-922F-0778314DDAF0")]
[SiteImageUrl]
[Access(Roles = "SiteA")]
public class ButtonBlock : SiteBlockData
{
    [Display(Order = 1, GroupName = SystemTabNames.Content)]
    [Required]
    public virtual string ButtonText { get; set; }

    [Display(Order = 2, GroupName = SystemTabNames.Content)]
    [Required]
    public virtual Url ButtonLink { get; set; }
}

1 public class SiteVirtualRole : VirtualRoleProviderBase
2 {
3     public string Site { get; set; }
4     public override void Initialize(string name, NameValueCollection config)
5     {
6         base.Initialize(name, config);
7         Site = config["site"];
8     }
9     public override bool IsInVirtualRole(IPrincipal principal, object context)
10    {
11        var site = HttpContext.Current?.Request?.Url?.Host;
12        if (string.IsNullOrEmpty(site) && (site.ToLower() == Site.ToLower())) return true;
13        return false;
14    }
15 }
```

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

Optimizely (Episerver) C# Tips and Tricks

