



EPISERVER AI CMS

Episerver Advance Recommendations on CodeArt

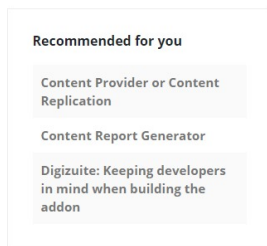


Allan Thraen | 7 Months Ago | PDF | 1



Some times you have so much great content on your website that you just wish you had a librarian to let your visitors know what to read next. And with Episerver Advance (Content Recommendations) you can at least have something that comes pretty close. I have been lucky enough to try it out on my site.

There are many approaches to doing content recommendations - and I'll probably do a number of blog posts on this topic as content+AI is one of my favorite topics. In the past I've done it using search fairly efficiently, I've done it using datamined Google Analytics (a great approach I might share in a future post) but of course there is also the Ferrari-approach, using all the sophisticated AI that Episerver has put together and made easily available. And since I got sponsored the service I did just that. It's been online for a few months and you might have noticed it on the bottom of the sidebar for each blog post. It looks something like this:



With Episerver tracking and Insight configured, using Advance is pretty straight forward. The hard part is really to keep track of all the different connection keys you get for all these different services :-)

The sample code provided on [github](#) has an approach I found a bit strange at first. It's a regular block, with a model, a controller and a view - just as we know it - but the view then contains javascript calling back to a Web API Controller that in turn calls the recommendation service, fetches results, renders them and returns the html to be inserted. So basically an AJAX approach.

But after some experimentation I must admit that in this case it's probably the best approach as a slow external service won't cause the site to slow down or crash - as it's loaded afterwards and async.

So, I chose the same solution, but simplified a few places. I also added a view for the final output so I can easily update the look & feel should I suddenly get an urge to show of my design skills.

I won't bother you with the controller or model for the base block, as that's completely straightforward (yes, it could even be using the default controller). The view however that does the callback looks like this in my code:

```
@using Episerver.Core
@using Episerver.Web.Mvc.Html
@model AllanTech.Web.Models.ViewModels.AdvanceViewModel
@{ Layout = null; }

<div class="widget categories">
  <header>
    <h3 class="h6" @Html.EditAttributes(m => m.CurrentBlock.Name)@Model.CurrentBlock.>
  </header>
  <div class="advancerecs"></div>

  <script>
    $(function(){
      $.ajax('/Advance/?contentId=@Model.ContentId&numRecs=@Model.Current
        success:function (data) {
          $('#advancerecs').html(data);
        }
      });
    });
  </script>
</div>
```

Blog posts

2019

September

August

July

June

May

April

March

February

January

2018

December

November

October

September

August

June

Available Topics

.NET DEVELOPMENT

ADDON DEVELOPMENT

AI

API BUILDING

AZURE

BEHAVIOR ANALYTICS

BIG DATA

C#

CMS

CONTENTFUL

DIGIZUITE

ELASTICSEARCH

EPISERVER

FREELANCING

INFORMATION RETRIEVAL

INTEGRATIONS

LIFE AS A CODER

MICRO SERVICES

OFFSHORE DEVELOPMENT

PRODUCT MANAGEMENT

TECH TALK

TIPS AND TRICKS

VISION DEMOS & PROTOTYPES

About Allan

I'm a freelance software architect / developer and Episerver expert (EMVP) based in Copenhagen Denmark. I've worked for Episerver for more than 10 years before going freelance and understand both your business needs as well as the technical possibilities.



I love solving real business problems with innovative and creative coding!

Learn more about my skills and specialities, reach out to me at info@codeart.dk or use this form to contact me.



That, in turn, calls back to my recommendations controller which fetches recommendations from Advance:



```
public class AdvanceRecommendationsController : Controller
{
    private readonly IRecommendationService _recommendationService;

    public AdvanceRecommendationsController(IRecommendationService recommendationService)
    {
        _recommendationService = recommendationService;
    }
    // GET: AdvanceRecommendations
    [Route("advance")]
    public async Task<ActionResult> Index(string contentId, int numRecs=3)
    {
        var rRequest = new RecommendationRequest
        {
            siteId = SiteDefinition.Current.Id.ToString(),
            context = new Context { contentId = contentId, languageId = "en" },
            numberOfRecommendations = numRecs
        };
        var res = await _recommendationService.Get(this.HttpContext, rRequest);
        return View(res);
    }
}
```

Recommended for you

- CodeArt
- Getting more insight into Profile Store
- About

And this is then rendered and returned to the calling javascript through this very basic view:

```
@using EPiServer.Core
@using EPiServer.Web.Mvc.Html
@model IEnumerable<EPiServer.Personalization.CMS.Model.RecommendationResult>
@{ Layout = null; }
@Foreach (var m in Model)
{
    <div class="item d-flex justify-content-between">@Html.ContentLink(m.Content.ContentLink)
}
}
```

As you can see - it's pretty straight forward.

Cool, but does it actually keep visitors engaged on the site longer?

Well, I wish I had a clear statistic that indicated that since I introduced this feature the number of pages per visit has increased tremendously - but sadly I don't. I'm working on improving my analytics setup though to give better insights as to how often the Advance links keep visitors on the site, but the data isn't ready yet.

But try to check it out when you are browsing my blog and see if those blog posts match your interests. Drop a comment and let me know what you think.

- EPISERVER
- AI
- CMS

Previous Post: [On Page Edit for Images](#)

Next Post: [Distributed Content: Delivering an Episerver Web Experience with Contentful Content](#)

Post Comments (1)

CodeArt ApS
Frederiksdalsvej 198, 2830 Virum, Denmark
Phone: +45 26 13 66 96
CVR: 39680688



Tweets by athraen