ALLAN THRAEN | 15 years ago | PDF |

USING OPENID WITH EPISERVER CMS

OpenID is an awesome idea. Imagine, having a single sign-on from whatever provider you choose, and using that to log in to all your favorite websites. Many sites already allows you to use them as OpenID providers and many more will support /consume OpenID logins. Naturally I wanted to have an easy way to make OpenID work with EPiServer CMS. And what would be more natural than a Membership provider for OpenID?! At least, that was my initial thought. But after struggling with making one I realized it wasn't as easy as it sounded – mainly because the OpenID protocol relies heavily on client-side redirects.

There are many good guides with great explanations on how OpenID works. I'll just try to dumb it down so even I understand: When you want to use your OpenID identity to log in to any given website you provide it with your identity. That identity is a string that points to a webpage you "own" (could be your blog from blogpost, your profile page from facebook, etc). As an example I like to use my blogspot account: allantech.blogspot.com. The page you reach on it has a meta-tag like this:

<link rel="openid.server" href="http://www.blogger.com/openid-server.g" />

Knowing which server, will help the website I'm trying to log in to identify where it should redirect me. And that's just what it'll do. After I give it my identity it'll redirect me to that identity's login page. You might already be logged in to that service, in which case you'll continue straight to a page where the openid server will ask you if it can tell the website you are trying to access that you own that identity.

If I click "yes" it'll send me back to the original website, together with a token that'll tell the website that I in fact own the identity I claim to own. And that what it's all about – not really knowing who you are – but knowing that you are definitely a person allowed to use a given identity string.

There are some extensions to the protocol that'll allow the openid server to share more information with the website – but for simplicity I'll disregard that now.

Anyway – how could we go about doing this from an EPiServer installation?! After some contemplation I decided that a simple user control code sample would do the trick. Quick and easy – and if I later need to, I can easily convert it into Dynamic Content, using the DCPlugin. Since every user on a system can have multiple OpenID identities, we should also have a table in the database mapping OpenIDs to users – however, also that I have skipped in this sample for the sake of simplicity. Instead I have simply decided that your OpenID Identity = your episerver username. So this simplifies my user control quite a bit. If a user who's not logged in see's it, it should allow him/her to provide their Identiy and log in. If they provide an identity that doesn't exist, it should get some more details about them and create a new user with that identity. That's it.

For the OpenID protocol communication we are in luck – there's a couple good libraries for .NET. For this example I decided to go with ExtremeSwanks library. From thereon I mostly just customized their sample code:

```
using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using EPiServer;
using EPiServer.Core;
using EPiServer.DataAbstraction;
using EPiServer.Web.WebControls;
using ExtremeSwank.OpenId;
```

How do you use this sample

1. Download ExtremeSwanks OpenID library. Add a reference to the assembly in your project.
2. Download the sample user control.
3. Make sure you are using SqlServerMembership provider. And if you are using the Multiplexing provider, make sure SqlServerMembership is provider1 – so it allows you to create users.
4. Include the user control and code-behind in your project. Customize their look, feel and functionality.
5. Put the user control wherever you want the log in to be.

Code sample is released AS-IS. Use as you please. Copyright © 2009 EPiServer AB.

Enjoy!

P.S. Kudos to @tednyberg for tweeting me into blogging this today.