

Microsoft Active Directory als Group Provider (deprecated)

- [Introductie](#)
- [Methoden](#)
 - [Toon mijn eigen groepen](#)
 - [Toon alle leden van 1 van mijn groepen](#)
 - [Toon een bepaald persoon](#)
- [Authenticatie](#)
- [Todo](#)



Let op! De documentatie op deze pagina is nog niet aangepast naar de nieuwe omgeving. Op de nieuwe omgeving is het niet meer mogelijk om alle leden van een bepaalde groep te tonen. Dit endpoint hoef je dus niet meer te implementeren op je *group provider* systemen.

Introductie

Met de kennis van het VOOT-protocol in het achterhoofd heeft SURFnet een simpele webdienst geïmplementeerd om toegang te krijgen tot Microsoft Active Directory (AD) als Group Provider. SURFnet gebruikt hiervoor het LDAP-protocol. Voor de ontwikkeling van dit protocol is Microsoft Visual Studio 2010 gebruikt. Het project is gebaseerd op het 'WCF REST Services Project Template'.

Om verbinding te maken met de LDAP-server heb je de component [LDAP library for C#.NET](#) nodig. De AD-server wordt opgezet met de standaardconfiguratie, die de mappen 'Builtin' en 'Users' bevat.

Op dit moment ondersteunt deze tutorial alleen bepaalde delen van het VOOT-protocol.

Methoden

De onderstaande methoden worden op dit moment ondersteund:

- toon mijn eigen groepen - HTTP GET naar '/groups/@me'
- toon alle deelnemers van een van mijn eigen groepen - HTTP GET naar '/people/@me/[groep-id]'
- toon een bepaald persoon - HTTP GET naar '/people/[gebruiker-id]'

De code voor deze methoden is als volgt:

```

public class VootService
{
    private readonly IVootReader reader;

    public VootService()
    {
        reader = new LdapVootReader(new ConfigurationHelper());
    }

    [WebGet(UriTemplate = "groups/@me")]
    public string IsMemberOf()
    {
        return reader.IsMemberOf(currentUser);
    }

    [WebGet(UriTemplate = "people/{userId}")]
    public string GetMember(string userId)
    {
        return reader.GetMember(userId);
    }

    [WebGet(UriTemplate = "people/@me/{groupId}")]
    public string GetGroupMembers(string groupId)
    {
        return reader.GetGroupMembers(groupId);
    }
}

```

Alle resultaten worden geretourneerd in JSON-formaat.

Toon mijn eigen groepen

De 'VootReader'-klasse bevat de methode 'isMemberOf(userId)'. Deze retourneert alle groepen waar de gebruiker lid van is. De LDAP-bron wordt doorzocht vanuit 'CN=Users, DN=demo, ..., DN=nl' met het filter 'samAccountName=[gebruiker-id]'.

Het attribuut 'memberOf' retourneert een 'array' met alle groepen waar de gebruiker lid van is. Dat kan er bijvoorbeeld zo uitzien:

```

{
  "CN=TestGroup,CN=Users,DC=demo,DC=.,DC=.,DC=surfnet,DC=nl",
  "CN=DemoGroup,CN=Users,DC=demo,DC=.,DC=.,DC=surfnet,DC=nl",
  "CN=Administrators,CN=Builtin,DC=demo,DC=.,DC=.,DC=surfnet,DC=nl"
}

```

Alle elementen uit de 'array' worden verwerkt door het verkrijgen van de enige 'LDAP'-entry die gelijk is aan de eerste 'common name' (CN).

Toon alle leden van 1 van mijn groepen

De methode 'GetGroupMembers' retourneert alle leden van de opgegeven groep. De LDAP-zoektermen zijn gelijk aan de vorige methode, omdat in dit geval de groepen onderdeel zijn van de 'Users'-map. Maak gebruik van 'groupId' als deel van het filter.

Opnieuw wordt het resultaat geretourneerd als een 'array', maar ditmaal als waarde van het 'member'-attribuut.

Toon een bepaald persoon

De methode 'GetMember' retourneert de gegevens van de desbetreffende gebruiker. Het belangrijkste onderdeel van de methode is:

```
// Get the entry.
LdapEntry entry = GetEntry(id, configHelper.PeopleDn);

// Process the result.
if (entry != null)
{
    LdapAttribute attr = entry.getAttribute("displayName");
    if (attr != null)
    {
        result = new Person { DisplayName = attr.StringValue, Id = id };
    }
}
```

Authenticatie

Voor het authenticeren van het eindpunt van de gebruiker gebruiken we '3-legged OAuth versie 1.0a'. Om Single Sign On mogelijk te maken tussen de Service Provider, SURFconext en het externegroepen-eindpunt, moet je de OAuth Provider (deze is verantwoordelijk voor het aanmaken en uitdelen van toegangs-'tokens' na een succesvolle authenticatie) koppelen met SURFconext.

Hiervoor zijn verschillende .NET-componenten beschikbaar. SURFconext maakt gebruik van het component DotNetOpenAuth. Een voorbeeld hiervan is een 'OpenAuthServiceProvider'. Dit voorbeeld hebben we als basis gebruikt voor de OAuth/implementatie. Het bevat een ASP.NET handler door de 3-legged OAuth handshake. Vervolgens wordt er een VOOT-webservice aan toegevoegd. Met een aparte webapplicatie kun je de functionaliteit van de implementatie testen.

Todo

- Voeg extra VOOT-parameters toe: count, filterBy, filterOp, filterValue, sortOrder, startIndex.
- Voeg de code toe aan de GitHub directory