# Integration von Microsoft® .NET Framework / Core mit DoubleClue using OpenID

## 1. Einführung

Diese Anleitung ist für .NET Framework / Core Applications Entwickler gedacht, die OpenID / OAuth zur Authentifizierung verwenden und DoubleClue Multi-Factor Authentication in ihr Produkt integrieren wollen.

Anforderungen:

- .NET Framework-Anmeldung mit Microsoft.Owin Installation ODER .NET Core web application.
- DoubleClue Enterprise Management (DCEM) Installation mit registrierten Benutzern.

## 2. Vorbereitung von DCEM als OpenID Authentication Server

Um DCEM darauf vorzubereiten, ein Authentication Server zu werden, schlagen Sie bitte Kapitel 12 im “DCEM_Manual_DE.pdf” nach.

## 3. Registrierung der Anmeldung als OpenID-Nutzer für DCEM

1. Öffnen Sie “OpenID-OAuth” im DCEM Hauptmenü und anschließend das Untermenü “Client Metadata”.
2. Klicken Sie auf “Add”.
3. Geben Sie einen vielsagenden Anzeigenamen in das “Display Name“-Feld ein, um den Eintrag schnell wiederfinden zu können.
4. Geben Sie Ihre “Client ID” an, wenn Sie eine haben, oder klicken Sie auf “Generate”, um eines anzulegen.
5. Geben Sie Ihr “Client Secret” an oder klicken Sie auf “Generate”, um eines anzulegen.
6. Klicken Sie auf “OK”.

Ihre Anmeldung als OpenID-Client für DCEM ist jetzt eingetragen.

# 4. Verbindung von .NET Framework Web Application mit DCEM

Fügen Sie die folgenden Einstellungen zur Configuration-Method Ihrer OWIN Startup class hinzu.

```
app.UseOpenIdConnectAuthentication(new OpenIdConnectAuthenticationOptions
{
    ClientId = clientId,
    ClientSecret = clientSecret,
    Authority = tokenUri,
    RedirectUri = redirectUri,
    ResponseType = OpenIdConnectResponseType.CodeIdTokenToken,
    Scope = OpenIdConnectScope.OpenIdProfile,
    TokenValidationParameters = new TokenValidationParameters
    {
        IssuerSigningKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(clientSecret))
    },
    Notifications = new OpenIdConnectAuthenticationNotifications
    {
        AuthorizationCodeReceived = async n =>
        {
            HttpClient client = new HttpClient();

            // Get Access Token
            TokenResponse tokenResponse = await client.RequestAuthorizationCodeTokenAsync(
                new AuthorizationCodeTokenRequest
            {
                Address = tokenUri,
                ClientId = clientId,
                ClientSecret = clientSecret,
                RedirectUri = redirectUri,
                Code = n.Code
            });
            if (tokenResponse.IsError) throw new Exception(tokenResponse.Error);

            // Get User Claims
            UserInfoResponse userInfoResponse = await client.GetUserInfoAsync(new UserInfoRequest
            {
                Address = userInfoUri,
                Token = tokenResponse.AccessToken
            });
            if (userInfoResponse.IsError) throw new Exception(userInfoResponse.Error);

            n.AuthenticationTicket.Identity.AddClaims(userInfoResponse.Claims);
        },
    },
});
```

- ***clientId*** ist die Client ID, die in DCEM registriert wurde
- ***clientSecret*** ist das Client Secret, das in DCEM registriert wurde
- ***tokenUri*** ist die URL, die im Issuer Feld in der Einstellungenanzeige in DCEM's OpenID-OAuth Modul angezeigt wird zusammen mit "/dcem/oauth". Zum Beispiel, wenn der Aussteller https://dcem:8080 ist, ist die tokenUri "https://dcem:8080/dcem/oauth".
- ***userInfoUri*** ist die gleiche *tokenUri*, an die zusätzlich noch "/userinfo" am Ende hinzugefügt wird. Entsprechend unseres vorherigen Beispiels wäre die userInfoUri "https://dcem:8080/dcem/oauth/userinfo".
- ***redirectUri*** ist eine URL, die Sie unter Ihrer Domain, welche als Client angemeldet wurde, wählen können. OWIN verlangt diese URL, um die Location, von der es Authorisation Codes und Access Tokens, erhält finden zu können.

# 5. Verbindung einer .NET Core Web Application mit DCEM

Fügen Sie diese Einstellungen in die Configuration-Method der ASP.NET Core Startup class hinzu.

```
app.UseOpenIdConnectAuthentication(new OpenIdConnectOptions
{
    SaveTokens = true,
    ClientId = clientId,
    ClientSecret = clientSecret,
    ResponseType = OpenIdConnectResponseType.Code,
    TokenValidationParameters = new TokenValidationParameters
    {
        IssuerSigningKey = new
SymmetricSecurityKey(Encoding.UTF8.GetBytes(clientSecret))
    },
    Authority = tokenUri,
});
```

- ***clientId*** ist die Client ID, die in DCEM registriert wurde
- ***clientSecret*** ist das Client Secret, das in DCEM registriert wurde
- ***tokenUri*** ist die URL, die im Issuer Feld in der Einstellungenanzeige in DCEM's OpenID-OAuth Modul angezeigt wird zusammen mit "/dcem/oauth". Zum Beispiel, wenn der Aussteller https://dcem:8080 ist, ist die tokenUri "https://dcem:8080/dcem/oauth".
- ***userInfoUri*** ist die gleiche *tokenUri*, an die zusätzlich noch "/userinfo" am Ende hinzugefügt wird. Entsprechend unseres vorherigen Beispiels wäre die userInfoUri "https://dcem:8080/dcem/oauth/userinfo".