



# External Authentication IdentityServer (Duende)

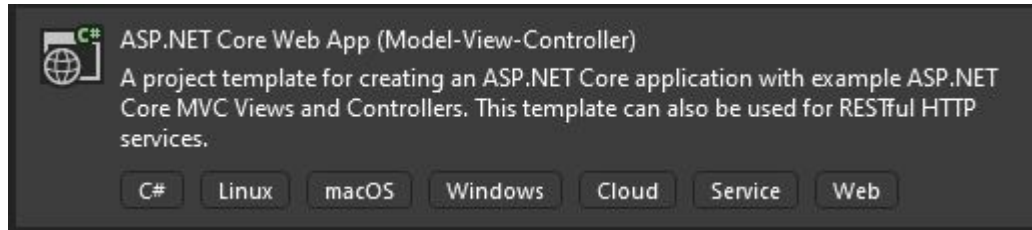
C# Web 2

**DE HOGESCHOOL  
MET HET NETWERK**

Hogeschool PXL – Elfde-Liniestraat 24 – B-3500 Hasselt  
[www.pxl.be](http://www.pxl.be) - [www.pxl.be/facebook](https://www.pxl.be/facebook)



# ASP.Net Core MVCAuthentication



# Authentication

## Views/Account

- Login
- FacebookLogin
- GoogleLogin
- Register

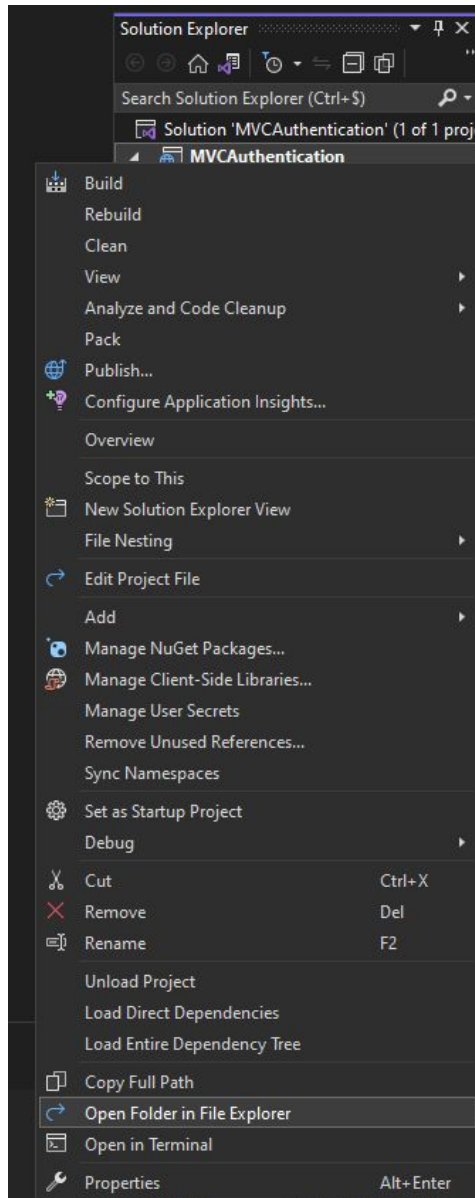
## Controller/AccountController

- Login
- FacebookLogin
- GoogleLogin
- Register
- Logout

# Authentication - overzicht

- HomeController
  - Geen authorisatie
  - Indien development
  - Overzicht van Rollen en gebruikers
  - Aanmaken van Admin user
- TestController
  - Authorisatie
    - Register
    - Login

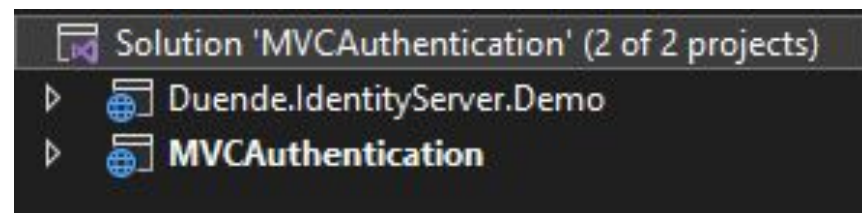
# External Authentication - IdentityServer



- Copy project file naar solution folder
  - DuendelIdentityServer.zip (Blackboard)

Naam	Gewijzigd op	Type	Grootte
.vs	3/03/2023 18:56	Bestandsmap	
DuendelIdentityServer	10/03/2023 11:46	Bestandsmap	
MVCAuthentication	10/03/2023 9:54	Bestandsmap	
DuendelIdentityServer.zip	10/03/2023 11:45	Gecomprimeerde ...	2.032 kB
MVCAuthentication.sln	3/03/2023 11:35	Visual Studio Solu...	2 kB

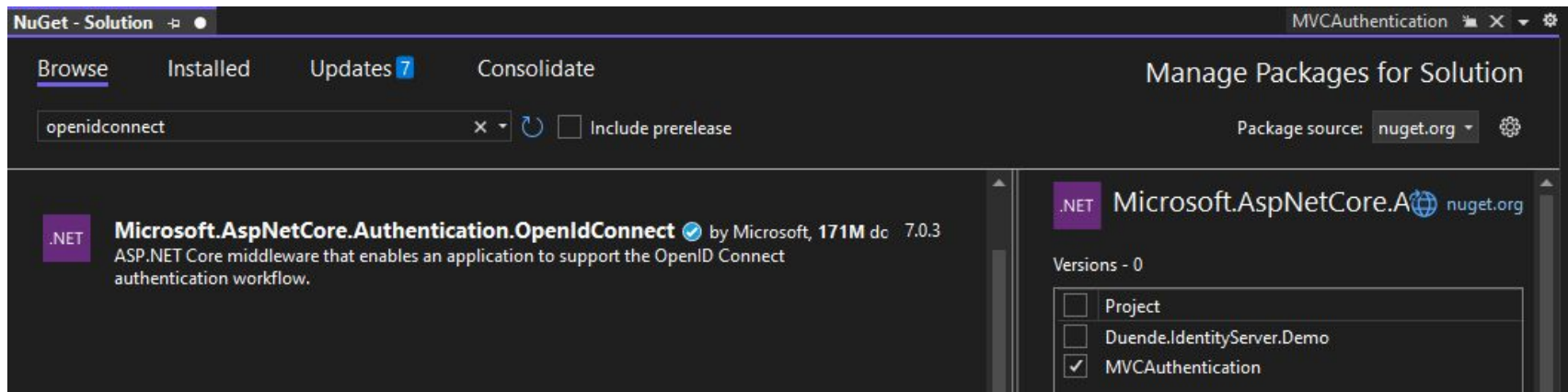
- Visual Studio – Solution explorer
  - Solution – right click – Add existing project
    - DuendelIdentityServer



# External Authentication – IdentityServer Nuget package

Toevoegen aan MVCAuthentication project:

- Microsoft.AspNetCore.Authentication.OpenIdConnect



# External Authentication – IdentityServer

## Poortnummers

### Poortnummer Duende server (Duende.IdentityServer.Demo):

- Zie Properties/launchSettings.json
- "applicationUrl": "https://localhost:5001;http://localhost:5000",

### Poortnummer applicatie (MVCAuthentication):

- Zie Properties/launchSettings.json
- "applicationUrl": "https://localhost:7001;http://localhost:7000",

Deze moeten verschillend zijn!!!

# External Authentication – IdentityServer

## Multiple Startup Projects

- **Multiple startup projects**
  - Eerst Duende server opstarten en dan pas de MVC applicatie.



# External Authentication – IdentityServer Program.cs

## Program.cs

```
...
builder.Services.AddAuthentication()
    .AddCookie("Cookies")
    .AddOpenIdConnect("oidc", options =>
    {
        options.Authority = "https://localhost:5001";
        options.ClientId = "mvc";
        options.ClientSecret = "secret";
        options.ResponseType = "code";
        options.SaveTokens = true;
        options.Scope.Add("openid");
        options.Scope.Add("profile");
        options.Scope.Add("email");
        options.GetClaimsFromUserInfoEndpoint = true;
    });
...
```

# Login

LoginUser

---

UserName

Password

Login

Login Facebook Login Google Login Duende

## Views/Account/Login.cshtml

```
<div class="form-group m-1 p-1">
  <input type="submit" value="Login" class="btn btn-primary" />
</div>
<!-- Facebook -->
<!-- Google -->
<div class="btn btn-dark m-1 p-1 text-white">
  <a class="link-light" asp-controller="Account"
    asp-action="DuendeLogin">Login Duende</a>
</div>
```

# External Authentication

## Duende

### Views/Account/login

- Add Button – Action: DuendeLogin

### AccountController

```
public IActionResult DuendeLogin()
{
    string returnUrl = Url.Action("DuendeResponse", "Account");
    string scheme = "oidc";
    var properties = _signInManager.ConfigureExternalAuthenticationProperties(
        scheme, returnUrl);
    return new ChallengeResult(scheme, properties);
}
```



Login

LoginUser

UserName

Password

Login

Login Facebook Login Google Login Duende

# External Authentication - Duende

## AccountController - DuendeResponse

```
public async Task<IActionResult> DuendeResponse()
{
    ExternalLoginInfo externalLoginInfo =
        await _signInManager.GetExternalLoginInfoAsync();
    if (externalLoginInfo == null)
    {
        return RedirectToAction(nameof(Login));
    }

    // ...
    return View(model);
}
```

# External Authentication - Duende

## AccountController - DuendeResponse

-> Kijk in Debug naar de claim informatie, is iets anders dan Google & Facebook

```
public async Task<IActionResult> DuendeResponse()
{
    // ...
    string userName = externalLoginInfo.Principal.FindFirst("name").Value;
    UserViewModel model = new UserViewModel()
    {
        UserName = userName,
        Email = externalLoginInfo.Principal.FindFirst("email").Value
    };
    // ...
    return View(model);
}
```

# External Authentication - Duende

## AccountController - DuendeResponse

```
public async Task<IActionResult> DuendeResponse()
{
    // ...
    Microsoft.AspNetCore.Identity.SignInResult result =
        await _signInManager.ExternalLoginSignInAsync(
            externalLoginInfo.LoginProvider, externalLoginInfo.ProviderKey, false);
    if (!result.Succeeded)
    {
        var identityResult = await CreateIdentityUserAsync(
            externalLoginInfo, model);
        if (!identityResult.Succeeded)
        {
            return View("Login");
        }
    }
    // ...

    return View(model);
}
```

# External Authentication - Duende

AccountController - DuendeResponse (→ verschilt tov Google- en FacebookResponse !!!)

- `private IdentityUser GetIdentityUser(ExternalLoginInfo info, UserViewModel model)`
- `private async Task<IdentityResult> CreateIdentityUserAsync(ExternalLoginInfo info, UserViewModel model)`

Extra parameter!

# External Authentication - Duende

GetIdentityUser:

```
private IdentityUser GetIdentityUser(
    ExternalLoginInfo info, UserModel model)
{
    //string userName = info.Principal.FindFirst("name").Value;
    string userName =
        $"{model.UserName}_{info.LoginProvider}_{info.ProviderKey}";
    //string email = info.Principal.FindFirst("email").Value;
    IdentityUser user = new IdentityUser(userName)
    {
        Email = model.Email
    };
    return user;
}
```



# External Authentication - Duende

## CreateIdentityUserAsync:

```
private async Task<IdentityResult> CreateIdentityUserAsync(
    ExternalLoginInfo externalLoginInfo, UserViewModel model)
{
    // Enige verschil met Facebook / Google staat gearceerd in geel
    IdentityUser user = GetIdentityUser(externalLoginInfo, model);
    IdentityResult identityResult = await _userManager.CreateAsync(user);
    if (identityResult.Succeeded)
    {
        identityResult = await _userManager.AddLoginAsync(
            user, externalLoginInfo);
        if (identityResult.Succeeded)
        {
            await _signInManager.SignInAsync(user, false);
        }
        else
        {
            return IdentityResult.Failed(
                new IdentityError { Description = "error in AddLogin" });
        }
    }
    return identityResult;
}
```

## AccountController – DuendeResponse

```
public async Task<IActionResult> DuendeResponse()
{
    ExternalLoginInfo externalLoginInfo =
        await _signInManager.GetExternalLoginInfoAsync();
    if (externalLoginInfo == null)
    {
        return RedirectToAction(nameof(Login));
    }
    string userName = externalLoginInfo.Principal.FindFirst("name").Value;
    UserViewModel model = new UserViewModel()
    {
        UserName = userName,
        Email = externalLoginInfo.Principal.FindFirst("email").Value
    };
    Microsoft.AspNetCore.Identity.SignInResult result =
        await _signInManager.ExternalLoginSignInAsync(
            externalLoginInfo.LoginProvider, externalLoginInfo.ProviderKey, false);
    if (!result.Succeeded)
    {
        var identityResult = await CreateIdentityUserAsync(externalLoginInfo, model);
        if (!identityResult.Succeeded)
        {
            return View("Login");
        }
    }
    return View(model);
}
```

## ViewModels – UserViewModel

```
public class UserViewModel
{
    public string? Email { get; set; }
    public string Username { get; set; }
    public string Password { get; set; }
}
```

## Views – DuendeResponse

```
@model UserViewModel
```

```
@{
    ViewData["Title"] = "Duende response";
}
```

```
<h1 class="bg-info text-white">Your Login Details</h1>
```

```
<table class="table table-sm table-bordered">
```

```
    <tr>
```

```
        <th>Name</th>
```

```
        <th>Email</th>
```

```
    </tr>
```

```
    <tr>
```

```
        <td>@Model.Username</td>
```

```
        <td>@Model.Email</td>
```

```
    </tr>
```

```
</table>
```

[Home](#)   [Test authentication](#)

## Your Login Details

Name	Email
Kristof Palmaers	palmaerskristof@gmail.com

# Oefening IdentityServer Login

## Oefening

- Voeg nu aan de HrApp van vorig hoofdstuk de login via IdentityServer toe.
- Probeer te vermijden om alle authenticatie code rechtstreeks in de AccountController te schrijven. Maak voor de authenticatie code een aparte service met een eigen interface aan.

**Tip:** let vooral op de DuendeLogin en DuendeResponse methods, het SSL-poortnummer van de applicatie (bijvoorbeeld 7001, zie launchSettings.json) en ClientID en ClientSecret.