



Authentication

C# Web 2

DE HOGESCHOOL MET HET NETWERK

Hogeschool PXL – Elfde-Liniestraat 24 – B-3500 Hasselt
www.pxl.be - www.pxl.be/facebook





ASP.Net Core MVC Authentication

**GitHub Classroom Assignment:
MVC Authentication – Facebook**

<https://classroom.github.com/a/7CToZqNS>



ASP.Net Core MVCAuthentication

Additional information

ASP.NET Core Web App

C#

Linux

macOS

Windows

Cloud

Service

Web

Framework [i](#)

.NET 8.0 (Long Term Support)

Authentication type [i](#)

None

☒ Configure for HTTPS [i](#)

☐ Enable Docker [i](#)

Docker OS [i](#)

Linux

☐ Do not use top-level statements [i](#)



Packages



Microsoft.AspNetCore.Identity.EntityFrameworkCore by Microsoft
ASP.NET Core Identity provider that uses Entity Framework Core.



Microsoft.EntityFrameworkCore.SqlServer by Microsoft
Microsoft SQL Server database provider for Entity Framework Core.



Microsoft.EntityFrameworkCore.Tools by Microsoft
Entity Framework Core Tools for the NuGet Package Manager Console in Visual Studio.



ViewModels

```
namespace MVCAuthentication.ViewModels
{
    public class UserViewModel
    {
        public string? Email { get; set; }
        public string UserName { get; set; }
        public string Password { get; set; }
    }
}
```

Data

```
namespace MVCAuthentication.Data
{
    public class ApplicationDbContext : IdentityDbContext<IdentityUser>
    {
        public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options) :
base(options)
        {
        }
    }
}
```

appSettings.json

```
"ConnectionStrings": {
  "DefaultConnection": "Server=(localdb)\\mssqllocaldb;Database=MVCAuth;Trusted_Connection=True;
MultipleActiveResultSets=true" },
```

Program.cs

```
builder.Services.AddDbContext<ApplicationDbContext>(options =>
    options.UseSqlServer(builder.Configuration["ConnectionStrings:DefaultConnection"]));
builder.Services.AddIdentity<IdentityUser, IdentityRole>()
    .AddEntityFrameworkStores<ApplicationDbContext>();
```



Migration

NPM console

- Migrations
 - Initial
- Update-database



Authentication

Views/Account

- Login
- Register

Controller/AccountController

- Login
- Register
- Logout



Authentication - Register

```
public class AccountController : Controller
{
    UserManager<IdentityUser> _userManager;
    SignInManager<IdentityUser> _signInManager;
    public AccountController(UserManager<IdentityUser> userManager,
        SignInManager<IdentityUser> signInManager)
    {
        _userManager = userManager;
        _signInManager = signInManager;
    }

    [HttpGet]
    public IActionResult Register()
    { }

    [HttpPost]
    public async Task<IActionResult> RegisterAsync(UserViewModel model)
    { }
```

Authentication - Login

```
public AccountController(UserManager<IdentityUser> userManager,
    SignInManager<IdentityUser> signInManager)
{
    _userManager = userManager;
    _signInManager = signInManager;
}
...
[HttpGet]
public IActionResult Login()
{
    return View();
}
[HttpPost]
public async Task<IActionResult> LoginAsync(UserViewModel
model)
{
    return View();
}
```



Authentication - overzicht

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



ASP.NET MVC

EXTERNAL AUTHENTICATION

FACEBOOK SETTINGS

External Authentication

Facebook package



Microsoft.AspNetCore.Authentication.Facebook by Microsoft

ASP.NET Core middleware that enables an application to support Facebook's OAuth 2.0 authentication workflow.



External Authentication – Facebook

New App Settings

<https://developers.facebook.com/>

Indien nog geen developer account:

- Klik op Get Started.

Indien je al developer account hebt:

- Klik op My Apps > Create App.


Register

☐ Verify account

☐ Contact info

☐ About you

Welcome to Facebook for Developers



Create a Facebook for Developers account to build and manage apps that access the Facebook Graph API, contribute to apps that others own, and participate in the Facebook Developer community.

By proceeding, you agree to the [Facebook Platform Terms](#) and [Developer Policies](#)

Cancel Continue



External Authentication – Facebook

New App Settings

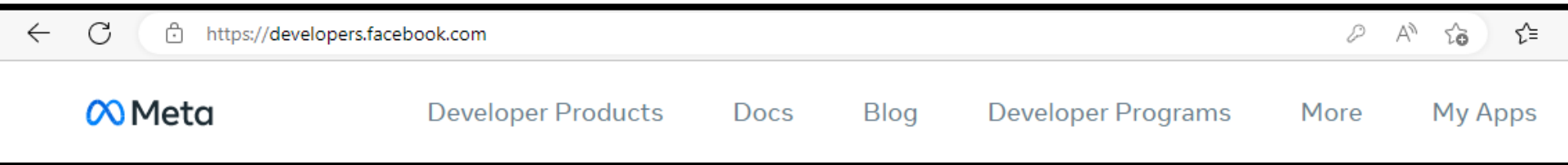
<https://developers.facebook.com/>

Indien nog geen developer account:

- Klik op Get Started.

Indien je al developer account hebt:

- Klik op My Apps





External Authentication – Facebook

New App Settings

☒ Register

☐ **Verify account**

☐ Contact info

☐ About you

Enter the Code from the SMS

Let us know this mobile number belongs to you. Enter the code in the SMS sent to [REDACTED] (Belgium).

[Send SMS Again](#)

Update Mobile Number

Continue



External Authentication – Facebook

New App Settings

✓ Register

✓ Verify account

Contact info

○ About you

Review Your Email Address

We use email addresses to send notifications, help you log in and personalize experiences, like connecting people and improving ads for everyone on our products. [Learn More](#)

Primary email

testpxl2020@gmail.com

☐ I agree to receive marketing-related electronic communications from Facebook, including developer news, updates and promotional emails. (You may unsubscribe from these emails at any time by clicking unsubscribe at the bottom of the email. You can also update your email preferences in Developer Settings.)

Update Email

Confirm Email



External Authentication – Facebook

New App Settings

✓ Register

✓ Verify account

✓ Contact info

About you

Which of the following best describes you?

Help us improve your experience by telling us which of the following roles best describe you.

</>

Developer

☐

Marketer

☐

Analyst

☐

Product Manager

☐

Student

☒

Owner/Founder

☐

Other

☐

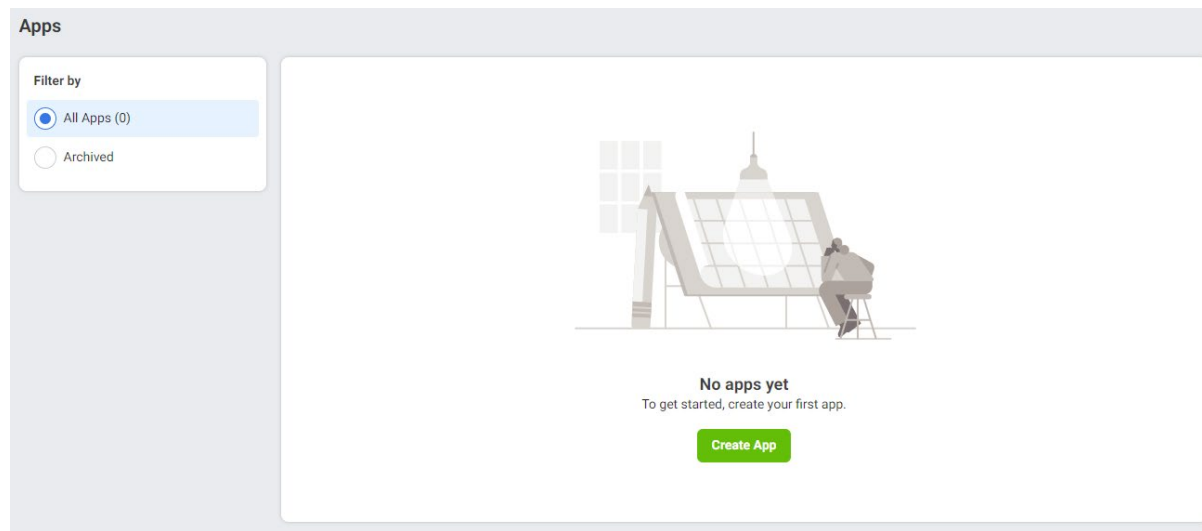
Complete Registration

18



External Authentication – Facebook

New App Settings





External Authentication – Facebook

New App Settings

Meta

DocsToolsSupportAppsRequired actions

Search

Create an app

☒ App details ☐ Use cases ☐ Business ☐ Finalize

App name

This is the app name that will show on your My Apps page and associated with your app ID. You can change the name later in Settings.

 4/30

App contact email

This is the email address we'll use to contact you about your app. Make sure it is an address you check regularly. We may contact you about policies, app restrictions or recovery if your app is deleted or compromised.

CancelNext



External Authentication – Facebook

New App Settings

Create an app

☒ App details

☒ Use cases

☐ Business

☐ Finalize

Add use cases

Some use cases can't be combined on the same app. Create another app if you need to add use cases that aren't combinable.

☐

Run ads to promote your app
Get an app ID to promote your app on Meta technologies, like Facebook and Instagram. [Learn more.](#)

☐

Access the Threads API
Use the Threads API and choose to authenticate users, retrieve user information, post threads, reply to threads, manage reply settings and/or gather insights for a Threads profile you own or manage on behalf of others. [Learn more.](#)

☐

Embed Threads content in other websites
Use the Threads oEmbed API to embed content of Threads posts, such as photos and videos, in other websites. [Learn more.](#)

☐

Launch a game on Facebook
Launch a game that players can find and play directly in their Feed or messages/conversations, on both desktop and mobile devices. [Learn more.](#)

☒

Authenticate and request data from users with Facebook Login
Our most common use case. A secure, fast way for users to log into your app or game and for the app to ask for permissions to access their data to personalize their experience. [Learn more.](#)

Looking for something else?

If you need something that isn't shown above, you can see more options by selecting Other.

☐

Other
Explore other products and data permissions such as ads management, WhatsApp and more. You'll be asked to select an app type and then you can add the permissions and products you need.

Cancel

Previous

Next



External Authentication – Facebook

New App Settings

Create an app

☒ App details

☒ Use cases

☐ Business

☐ Finalize

Which business portfolio do you want to connect to this app?

Connect a verified business portfolio to your app to get access to third-party user and business data from other business portfolios and publish this app. You can connect an unverified business portfolio or choose to add one later, but will be required to complete verification to gain data access.

☒ I don't want to connect a business portfolio yet.

Cancel

Previous

Next



External Authentication – Facebook

New App Settings

Create an app

✓ App details


✓ Use cases

⌵ Business


○ Finalize

Finalize

Requirements to publish app and maintain data access

 **Business verification**
Verify your business as a business entity in order to access user data through Meta's APIs.
[Learn more](#)

Required for 1 use case

 **App Review**
Complete all required data usage, data handling and data protection questions, then submit for App Review. [Learn more](#)

Required for 1 use case

By proceeding, you agree to the [Meta Platform Terms](#) and [Developer Policies](#).

Cancel

Previous

Go to dashboard



External Authentication – Facebook

New App Settings

2PRO

Dashboard

Required actions

Use cases

Review

Publish Unpublished

App ID

916448960568857

App secret

..... S...

Display name

2PRO

Namespace

App domains

Contact email

testpxl2020@gmail.com

Privacy Policy URL

http://www.pxl.be/

Terms of Service URL

http://www.pxl.be/

User data deletion

Data deletion instructions URL

http://www.pxl.be/

App icon (1024 x 1024)

Category

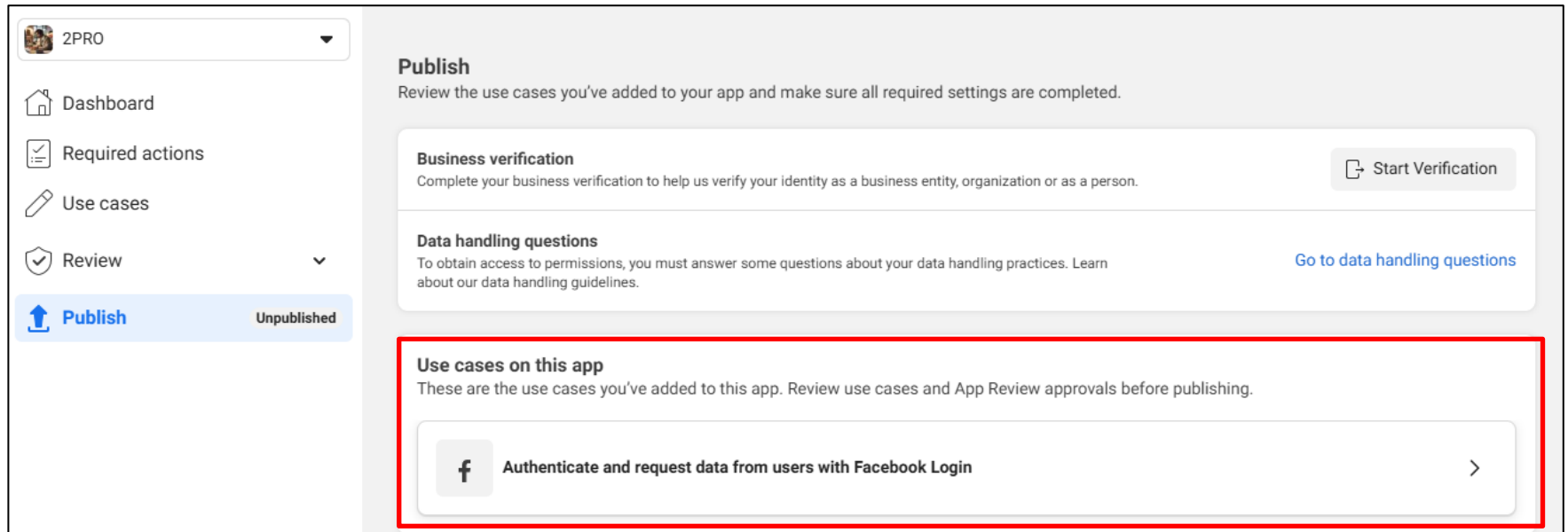
Education

Nodig in Program.cs



External Authentication – Facebook

New App Settings




The screenshot shows the 'Publish' section of the Facebook App Settings interface. On the left is a sidebar with navigation options: 'Dashboard', 'Required actions', 'Use cases', 'Review', and 'Publish' (which is highlighted in blue and labeled 'Unpublished'). The main content area is titled 'Publish' and contains three sections: 'Business verification' with a 'Start Verification' button, 'Data handling questions' with a link to 'Go to data handling questions', and 'Use cases on this app'. The 'Use cases on this app' section is highlighted with a red border and contains a single use case: 'Authenticate and request data from users with Facebook Login', which includes a Facebook icon and a right-pointing arrow.

Publish
Review the use cases you've added to your app and make sure all required settings are completed.

Business verification
Complete your business verification to help us verify your identity as a business entity, organization or as a person. [Start Verification](#)

Data handling questions
To obtain access to permissions, you must answer some questions about your data handling practices. [Learn about our data handling guidelines.](#) [Go to data handling questions](#)

Use cases on this app
These are the use cases you've added to this app. Review use cases and App Review approvals before publishing.

 **Authenticate and request data from users with Facebook Login** [>](#)

External Authentication – Facebook

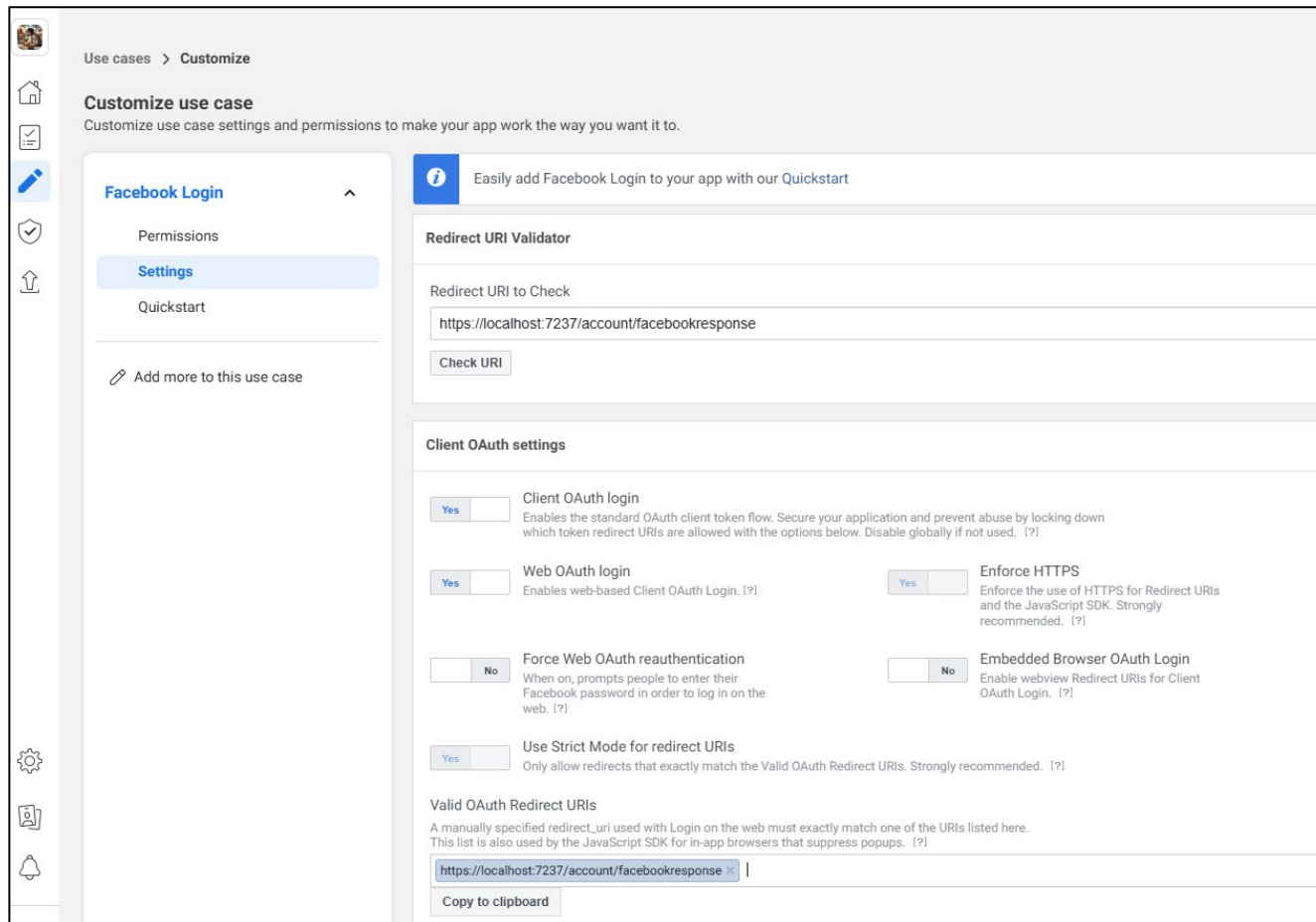
New App Settings

The screenshot shows the 'Customize use case' interface for 'Facebook Login'. On the left, a sidebar contains icons for home, checklist, edit, shield, and upload. The main content area has a header 'Use cases > Customize' and a sub-header 'Customize use case' with the instruction 'Customize use case settings and permissions to make your app work the way you want it to.' Below this, a left-hand menu lists 'Facebook Login', 'Permissions' (selected), 'Settings', and 'Quickstart'. The 'Permissions' section displays a table of permissions. The 'email' permission is highlighted with a red box. Below it, the 'public_profile' permission is listed.

Permissions ↑	API Calls	Info ⓘ ↑↓	Status ↑↓	Action
email The email permission allows your app to read a person's primary email address. Full Description ⓘ Requirements ⓘ	—	—		Add
public_profile The public_profile permission allows an app to read the Default Public Profile Fields on the User node. This permission is automatically granted to all apps. Full Description ⓘ Requirements ⓘ	0		Ready for testing	Delete

External Authentication – Facebook

New App Settings



Use cases > Customize

Customize use case

Customize use case settings and permissions to make your app work the way you want it to.

Facebook Login

- Permissions
- Settings**
- Quickstart

[Add more to this use case](#)

Redirect URI Validator

Redirect URI to Check

[Check URI](#)

Client OAuth settings

☒ **Client OAuth login**
Enables the standard OAuth client token flow. Secure your application and prevent abuse by locking down which token redirect URIs are allowed with the options below. Disable globally if not used. [?]

☒ **Web OAuth login**
Enables web-based Client OAuth Login. [?]

☐ **No** **Force Web OAuth reauthentication**
When on, prompts people to enter their Facebook password in order to log in on the web. [?]

☒ **Use Strict Mode for redirect URIs**
Only allow redirects that exactly match the Valid OAuth Redirect URIs. Strongly recommended. [?]

☒ **Enforce HTTPS**
Enforce the use of HTTPS for Redirect URIs and the JavaScript SDK. Strongly recommended. [?]

☐ **No** **Embedded Browser OAuth Login**
Enable webview Redirect URIs for Client OAuth Login. [?]


Valid OAuth Redirect URIs
A manually specified redirect_uri used with Login on the web must exactly match one of the URIs listed here. This list is also used by the JavaScript SDK for in-app browsers that suppress popups. [?]

[Copy to clipboard](#)



External Authentication – Facebook

New App Settings

 Meta

DocsToolsSupportApps

Apps

Filter by


☒ All Apps (8)

☐ Archived


☐ Required actions

Business portfolio [Clear](#)

No business portfolio selected ▼



2PRO
App ID: 916448960568857
Mode: In development

 Administrator

...



External Authentication – Facebook Flow

- Browser stuurt http request naar */Account/FacebookLogin*
- MVC app herleidt (redirect) de browser naar de facebook app
- Browser interageert met de facebook app (gebruiker bewijst wie hij of zij is)
- Facebook app herleidt (redirect) de browser naar de MVC app (*/signin-facebook?state=...*) met in de url een code waarmee de gegevens van de facebook gebruiker opgehaald kunnen worden
 - MVC middleware verwerkt de request naar */signin-facebook* door met de code, de appId en de appSecret bij facebook de gegevens van de gebruiker op te halen (token). Vervolgens wordt de gebruiker herleidt (redirect) naar */Account/FacebookResponse*
 - De MVC *facebookResponse* action methode van de *Account* controller verwerkt de data (token) ontvangen van facebook
 - Maakt een lokale gebruiker aan in de database (als deze nog niet bestaat)
 - Koppelt de id van de gebruiker bij facebook (provider key) aan de lokale gebruiker
 - Meldt de gebruiker aan (authentication cookie)

External Authentication – Facebook

Program.cs

(AllowUserNameCharacters)

Standaard laat het Identity framework geen username met spatie toe. Daarom gaan we de Identityoptions wijzigen in vorige slide.

facebook stuurt standaard “voornaam achternaam” als username terug.

Een tweede oplossing zou zijn om het emailadres zowel als email en als username te stockeren in onze tabel dan is deze controle overbodig.

De facebook user wordt opgeslagen in een aparte tabel.

- AspNetUserLogins

Maar ook is er een record toegevoegd aan de “AspNetUsers” tabel waar alle gebruikers worden opgeslagen (de externe facebook login wordt dus gekoppeld aan de AspNetUser via de AspNetUserLogins tabel).

External Authentication – Facebook

Program.cs

Services configureren - Program.cs

```
...
builder.Services.Configure<IdentityOptions>(options =>
{
    options.User.AllowedUserNameCharacters =
"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789-._@+ ";
    options.User.RequireUniqueEmail = true;
});
builder.Services.AddAuthentication()
    .AddFacebook(fbOpts=>
    {
        fbOpts.AppId = "911973619631699";
        fbOpts.AppSecret = "2d81d76565bc538b1ad3ac562a0281e1";
    });
```



External Authentication – Facebook Login

Views/Account/login

- Add Button – Action: FacebookLogin

AccountController

```
public IActionResult FacebookLogin()
{
    string returnUrl = Url.Action("FacebookResponse", "Account");
    var properties =
        _signInManager.ConfigureExternalAuthenticationProperties("Facebook", returnUrl);
    return new ChallengeResult("Facebook", properties);
}
```

Login

LoginUser

UserName

Password

Login

[Login via facebook](#)

[Registreer een nieuwe gebruiker](#)



AccountController - FacebookResponse

```
public async Task<IActionResult> FacebookResponse()
{
    //retrieve information that was send in the http request (by facebook)
    ExternalLoginInfo externalLoginInfo =
        await _signInManager.GetExternalLoginInfoAsync();

    if (externalLoginInfo == null)
    {
        //user did not login properly with facebook -> redirect to login page
        return RedirectToAction(nameof(Login));
    }

    ...

    return View();
}
```



AccountController - FacebookResponse

```
public async Task<IActionResult> FacebookResponse()
{
    ...
    //Put info provided by facebook (claims) into a viewmodel
    string userName = externalLoginInfo.Principal.FindFirst(ClaimTypes.Name).Value;
    //make sure username is unique
    UserViewModel model = new UserViewModel()
    {
        UserName = userName,
        Email = externalLoginInfo.Principal.FindFirst(ClaimTypes.Email).Value
    };

    ...

    return View(model);
}
```



AccountController - FacebookResponse

```
public async Task<IActionResult> FacebookResponse()
{
    ...
    //try to sign in with facebook user id (ProviderKey)
    Microsoft.AspNetCore.Identity.SignInResult result =
        await _signInManager.ExternalLoginSignInAsync(
            externalLoginInfo.LoginProvider, externalLoginInfo.ProviderKey, false);
    if (!result.Succeeded)
    {
        return View("Login");
    }
    return View(model);
}
```



AccountController - FacebookResponse

```
public async Task<IActionResult> FacebookResponse()
{
    ...
    //try to sign in with facebook user id (ProviderKey)
    Microsoft.AspNetCore.Identity.SignInResult result =
        await _signInManager.ExternalLoginSignInAsync(
            externalLoginInfo.LoginProvider, externalLoginInfo.ProviderKey, false);
    if (!result.Succeeded)
    {
        var identityResult = await CreateIdentityUserAsync(externalLoginInfo);
        if (!identityResult.Succeeded)
        {
            return View("login");
        }
    }
    return View(model);
}
```



AccountController - FacebookResponse

```
private IdentityUser GetIdentityUser(ExternalLoginInfo info)
{
    string userName = info.Principal.FindFirst(ClaimTypes.Name).Value;
    userName = $"{userName}_{info.LoginProvider}_{info.ProviderKey}";
    string email = info.Principal.FindFirst(ClaimTypes.Email).Value;
    IdentityUser user = new IdentityUser(userName)
    {
        Email = email
    };
    return user;
}
```

AccountController - FacebookResponse

```
private async Task<IdentityResult> CreateIdentityUserAsync(
    ExternalLoginInfo externalLoginInfo)
{
    //Put info provided by external provider (claims) into a viewmodel
    //Sign in failed -> user does not exist yet in our database -> create one
    IdentityUser user = GetIdentityUser(externalLoginInfo);
    IdentityResult identityResult = await _userManager.CreateAsync(user);
    if (identityResult.Succeeded)
    {
        //link the created user to the facebook login info
        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 - FacebookResponse

```
public async Task<IActionResult> FacebookResponse()
{
    //retrieve information that was send in the http request (by facebook)
    ExternalLoginInfo externalLoginInfo = await _signInManager.GetExternalLoginInfoAsync();
    if (externalLoginInfo == null)
    {
        //user did not login properly with facebook -> redirect to login page
        return RedirectToAction(nameof(Login));
    }
    string userName = externalLoginInfo.Principal.FindFirst(ClaimTypes.Name).Value;
    //make sure username is unique
    UserViewModel model = new UserViewModel()
    {
        UserName = userName,
        Email = externalLoginInfo.Principal.FindFirst(ClaimTypes.Email).Value
    };
    //try to sign in with facebook user id (ProviderKey)
    Microsoft.AspNetCore.Identity.SignInResult result =
        await _signInManager.ExternalLoginSignInAsync(
            externalLoginInfo.LoginProvider, externalLoginInfo.ProviderKey, false);
    if (!result.Succeeded)
    {
        var identityResult = await CreateIdentityUserAsync(externalLoginInfo);
        if (!identityResult.Succeeded)
        {
            return View("login");
        }
    }
    return View(model);
}
```



ViewModels – UserInfoModel

```
public class UserViewModel
{
    public string UserName { get; set; }
    public string Email { get; set; }
}
```

Views – FacebookResponse

```
@model UserViewModel
```

```
@{
    ViewData["Title"] = "Facebook 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>
```




Oefening Facebook Login

Oefening (Blackboard – Oefening 03)

- Voeg nu aan de HrApp van vorig hoofdstuk de login via Facebook 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 FacebookLogin en FacebookResponse methods!

Gebruik het juiste SSL-poortnummer van de applicatie (bijvoorbeeld 44379, zie launchSettings.json)

Gebruik de juiste facebook AppID en AppSecret!