



Google Authentication

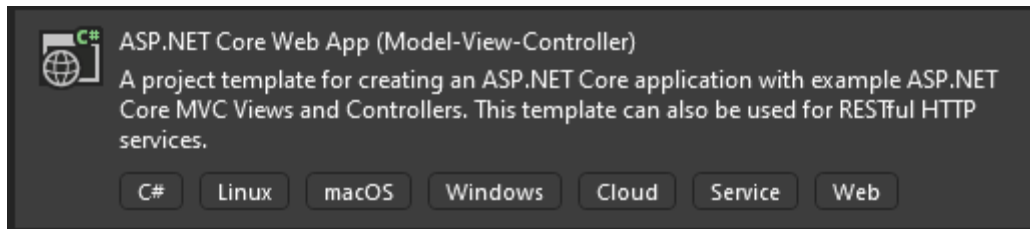
C# Web 2

**DE HOGESCHOOL
MET HET NETWERK**

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



ASP.Net Core MVCAuthentication



ViewModels

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

Authentication

Views/Account

- Login
- FacebookLogin
- Register

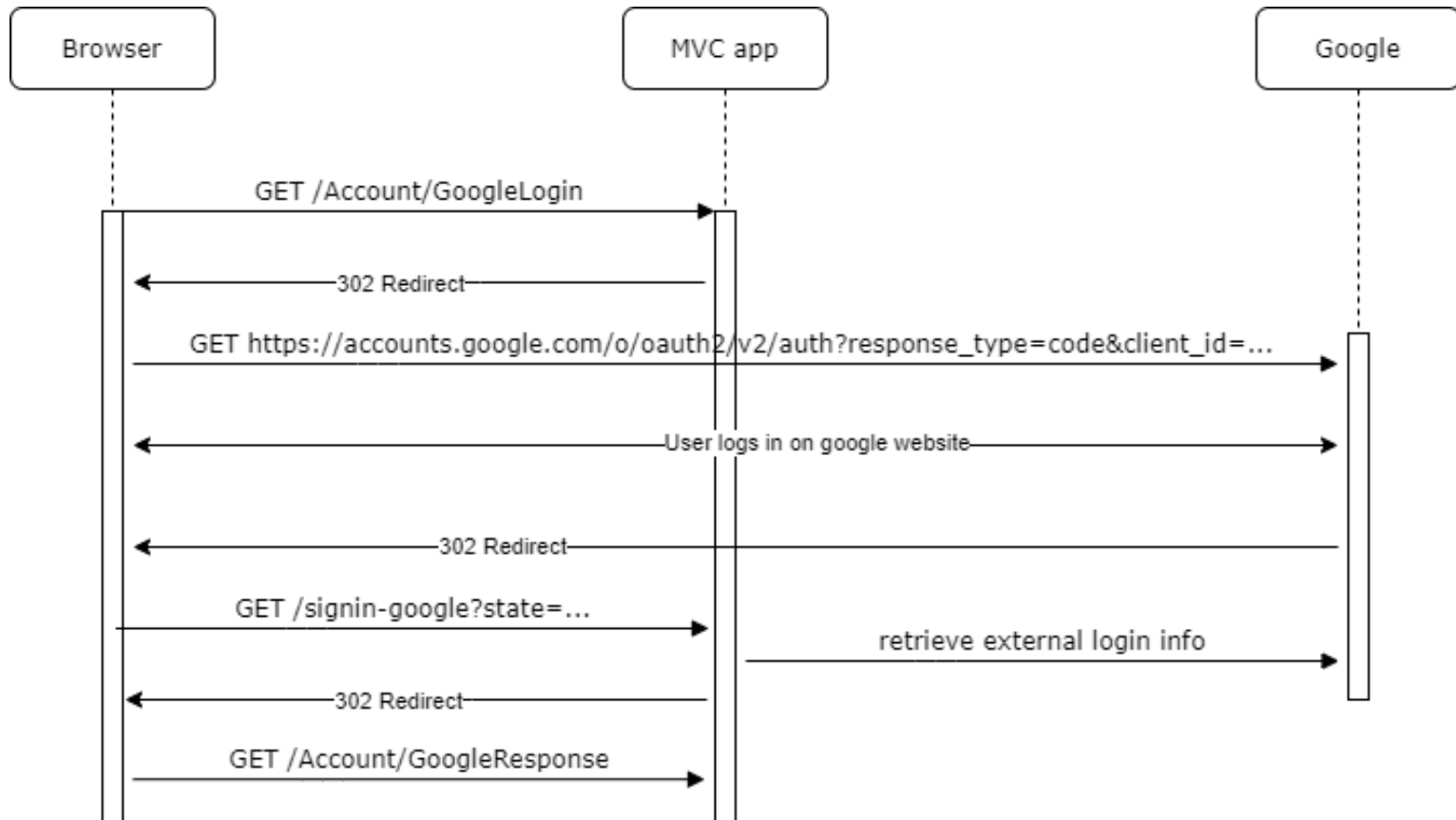
Controller/AccountController

- Login
- FacebookLogin
- Register
- Logout

Authentication - overzicht

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

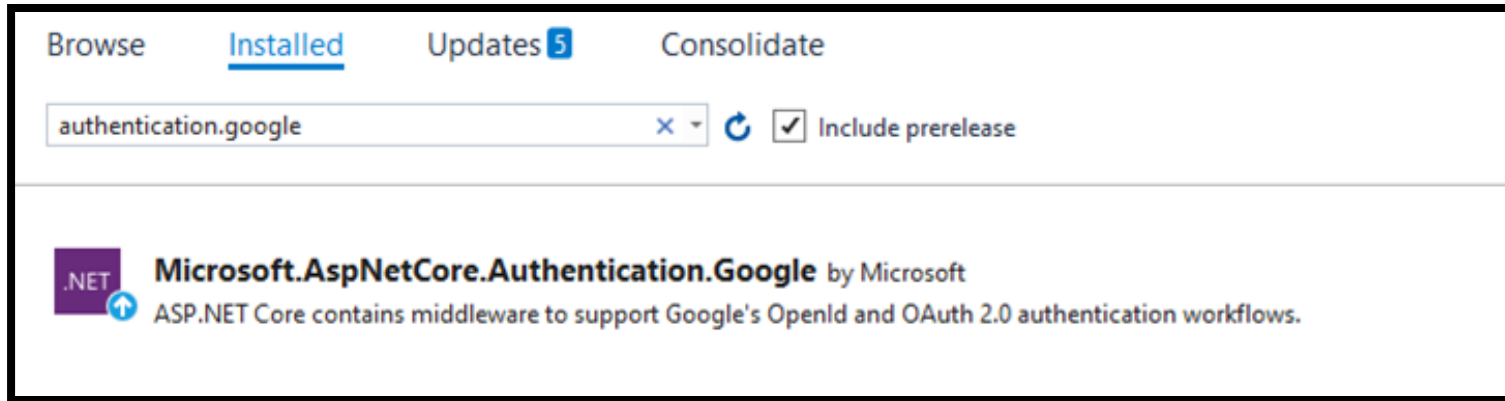
External Authentication – Google Flow



External Authentication – Google Flow

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

External Authentication - Google



<https://console.cloud.google.com/apis>
Login with your google account

The screenshot shows the Google Cloud Platform API console. The top navigation bar is blue with the Google Cloud Platform logo, a Google Login button, and a search bar. The left sidebar contains a menu with options: Dashboard, Bibliotheek, Inloggegevens (highlighted), OAuth-toestemmingsscherm, Domeinverificatie, and Pagina met gebruiksovereen... The main content area is titled 'Inloggegevens' and features a red-bordered button '+ INLOGGEGEVENS MAKEN' and a 'VERWIJDEREN' button. Below this, there is a section for 'API-sleutels' with a table header 'Naam' and 'Aanmaakdatum', and a message 'Er zijn geen API-sleutels om weer te geven'. Another section for 'OAuth 2.0-client-ID's' follows, with a table listing two client IDs: 'MVCGoogleHttps' and 'MVCGoogle', both created on '15 jan. 2021'.

Naam	Aanmaakdatum
MVCGoogleHttps	15 jan. 2021
MVCGoogle	15 jan. 2021

+ CREATE CREDENTIALS

DELETE

API key

Identifies your project using a simple API key to check quota and access

OAuth client ID

Requests user consent so your app can access the user's data

Service account

Enables server-to-server, app-level authentication using robot accounts

Help me choose

Asks a few questions to help you decide which type of credential to use

Google Cloud Platform

Google Login

Product

API

API's en services

Dashboard

Bibliotheek

Inloggegevens

OAuth-toestemmingsscherm

Domeinverificatie

Pagina met gebruiksovereen...

← ID voor OAuth-client maken

Een client-ID wordt gebruikt om een enkele app te identificeren op de OAuth-servers van Google. Als uw app op meerdere platforms draait, heeft elke app een eigen client-ID nodig. Bekijk [OAuth 2.0 instellen](#) voor meer informatie.

Type app *
Webapplicatie
[Meer informatie](#) over typen OAuth-clients

Naam *
MVCAuthentication

De naam van uw OAuth 2.0-client. Deze naam wordt alleen gebruikt om de client in de console te identificeren en wordt niet weergegeven aan eindgebruikers.

De domeinen van de URI's die u hieronder toevoegt, worden automatisch als [gemachtigde domeinen](#) toegevoegd aan uw [OAuth-toestemmingsscherm](#).

Geautoriseerde JavaScript-bronnen ?

Voor gebruik met verzoeken uit een browser

+ URI TOEVOEGEN

Geautoriseerde omleidings-URI's ?

Voor gebruik met verzoeken vanaf een webserver.

+ URI TOEVOEGEN

MAKEN

ANNULEREN

PXL

10

Name *

MVCAuthentication

The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.



The domains of the URIs you add below will be automatically added to your [OAuth consent screen](#) as [authorized domains](#).

Authorized JavaScript origins ?

For use with requests from a browser

+ ADD URI

Authorized redirect URIs ?

For use with requests from a web server

URIs

https://localhost:44302/signin-google

+ ADD URI

SAVE

CANCEL

OAuth client created

The client ID and secret can always be accessed from Credentials in APIs & Services



OAuth is limited to 100 [sensitive scope logins](#) until the [OAuth consent screen](#) is verified. This may require a verification process that can take several days.

Your Client ID

928731192351-19to95j33e764v2g1ub76ff81o2t1rgn.apps.g...



Your Client Secret

g4980ubdFvq2W0p0j0n_3085



Use the client id and client secret in your Startup.cs

OK

External Authentication – Google Program.cs

Program.cs

```
...
builder.Services.Configure<IdentityOptions>(options =>
{
    options.User.AllowedUserNameCharacters =
        "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789-._@+";
    options.User.RequireUniqueEmail = true;
});

builder.Services.AddAuthentication().AddGoogle(options =>
{
    options.ClientId =
        "840228473399-gencr6or0udsqckg55a1ras8ndf4bbqm.apps.googleusercontent.com";
    options.ClientSecret = "R9MI9BP_yYorCI-xEB4zMivB";
    options.SignInScheme = IdentityConstants.ExternalScheme;
});
...
```

External Authentication – Google

Program.cs

(AllowUserNameCharacters)

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

Google 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 google 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 google login wordt dus gekoppeld aan de AspNetUser via de AspNetUserLogins tabel).

Login

LoginUser

UserName

Password

Login

Login Facebook Login Google

[Registreer een nieuwe gebruiker](#)

Views/Account/Login.cshtml

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

External Authentication

GoogleLogin

Views/Account/login

- Add Button – Action: GoogleLogin

AccountController

```
public IActionResult GoogleLogin()
{
    string redirectUrl = Url.Action("GoogleResponse", "Account");
    var properties = _signInManager.ConfigureExternalAuthenticationProperties(
        "Google", redirectUrl);
    return new ChallengeResult("Google", properties);
}
```

Login

LoginUser

UserName

Password

Login

Login Facebook Login Google

[Registreer een nieuwe gebruiker](#)

External Authentication - Google

AccountController - GoogleResponse

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

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

    ...

    return View(); //goto googleResponse View
}
```


External Authentication - Google

AccountController - GoogleResponse

```
public async Task<IActionResult> FacebookResponse()
{
    ...
    //Put info provided by google (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();
}
```

External Authentication - Google

AccountController - GoogleResponse

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

External Authentication - Google

AccountController - GoogleResponse

```
public async Task<IActionResult> GoogleResponse()
{
    ...
    Microsoft.AspNetCore.Identity.SignInResult result =
        await _signInManager.ExternalLoginSignInAsync(
            externalLoginInfo.LoginProvider, externalLoginInfo.ProviderKey, false);
    if (result.Succeeded)
    {
        return RedirectToAction("Index", "Test");
    }
    else
    {
        var identityResult = await CreateIdentityUserAsync(externalLoginInfo);
        if (identityResult.Succeeded)
        {
            return RedirectToAction("Index", "Test");
        }
    }
    return View("login");
}
```

External Authentication - Google

AccountController - GoogleResponse (-> al aangemaakt voor FacebookResponse)

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

AccountController – GoogleResponse

```
public async Task<IActionResult> GoogleResponse()
{
    //retrieve information that was send in the http request (by google)
    ExternalLoginInfo externalLoginInfo =
    await _signInManager.GetExternalLoginInfoAsync();
    if (externalLoginInfo == null)
    {
        //user did not login properly with google -> redirect to login page
        return RedirectToAction(nameof(Login));
    }
    //Put info provided by google (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
    };
    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); //UserViewModel als model in GoogleResponse
}
```

ViewModels – UserViewModel

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

Views – GoogleResponse

```
@model UserViewModel
```

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

Login

LoginUser

UserName

Password

Login

google

[Registreer een nieuwe gebruiker](#)

[Home](#) [Test authentication](#)

Your Login Details

Name	Email
Kristof Palmaers	palmaerskristof@gmail.com

Oefening Google Login

Oefening

- Voeg nu aan de HrApp van vorig hoofdstuk de login via Google 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 GoogleLogin en GoogleResponse methods, het SSL-poortnummer van de applicatie (bijvoorbeeld 44379, zie launchSettings.json) en ClientID en ClientSecret.