

Laboratorium Security

Mieszko Kuczanski & Janek Baumgart

Wymagania

- Konto Azure
- Visual Studio
- Wszystko co wykorzystamy z tym laboratorium jest całkowicie darmowe.

Scenariusz

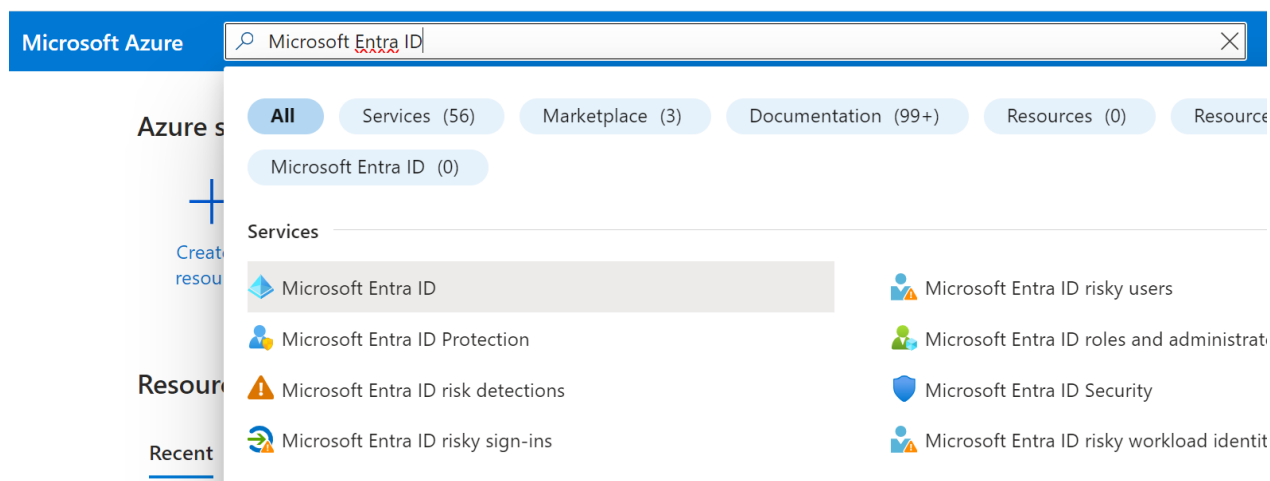
1. Omówienie podstaw

Zobaczmy jak zarejestrować aplikację w Microsoft Entra ID, przejdźmy przez podstawy.

1. Zalogujmy się do swojej subskrypcji platformy Azure

Z technicznego punktu widzenia wszystko, co zostanie pokazane nie wymaga subskrypcji platformy Azure. Można po prostu przejść do portalu.azure.com i zalogować się przy użyciu aktywnego konta i to zadziała. Dla ułatwienia i przejrzystości jednak skorzystamy z subskrypcji z których korzystałeś dotychczas.

2. Następnie wyszukaj „Microsoft Entra ID” i kliknij w odpowiedni kafelek aby dostać się do usługi Azure Active Directory (Microsoft Entra ID).



3. Jak już wybierzesz link i zostaniesz przeniesiony do Panelu Azure AD ujrzysz, że jest tu wiele rzeczy, ale my jesteśmy zainteresowani zarejestrowaniem aplikacji (App registrations), więc przechodzimy do rejestracji aplikacji.

4. Gdybyś miał dodatkowe aplikacje, które wcześniej zarejestrowałeś ukazały by się w liście. Nie mamy aktualnie żadnej, więc kliknijmy Nowa rejestracja i nadajmy jej nazwę.

Home > Default Directory >

Register an application

* Name

The user-facing display name for this application (this can be changed later).

Supported account types

Who can use this application or access this API?

☒ Accounts in this organizational directory only (Default Directory only - Single tenant)

☐ Accounts in any organizational directory (Any Azure AD directory - Multitenant)

☐ Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)

☐ Personal Microsoft accounts only

[Help me choose...](#)

Redirect URI (optional)

We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

[By proceeding, you agree to the Microsoft Platform Policies](#)

[Register](#)

5. Zwróćmy uwagę na wszystkie możliwe opcje. Może napisać wybrać aplikację tylko do naszego directory, do wielu różnych directory, wraz z osobistymi kontami Microsoft, lub tylko dla Kont Microsoft. My wybieramy pierwszą opcję – dla pojedynczego directory. Wybór ten można zmienić później. I klikamy zarejestruj.

Home > Default Directory >

AGHTEST

Search (Ctrl+F)

Overview

Quickstart

Integration assistant

Manage

Branding

Authentication

Certificates & secrets

Token configuration

API permissions

Expose an API

App roles

Owners

Roles and administrators | Preview

Manifest

Delete Endpoints Preview features

Got a second? We would love your feedback on Microsoft identity platform (previously Azure AD for developer). →

Essentials

Display name	: AGHTEST	Client credentials	: Add a certificate or secret
Application (client) ID	: 32fa9b42-bdd6-4caf-a178-f2dd106333c2	Redirect URIs	: Add a Redirect URI
Object ID	: 53a8780f-05fa-4140-9318-f90356df56b0	Application ID URI	: Add an Application ID URI
Directory (tenant) ID	: 47b926b3-3b6a-495d-8035-50e0f2fc002	Managed application in L.	: AGHTEST

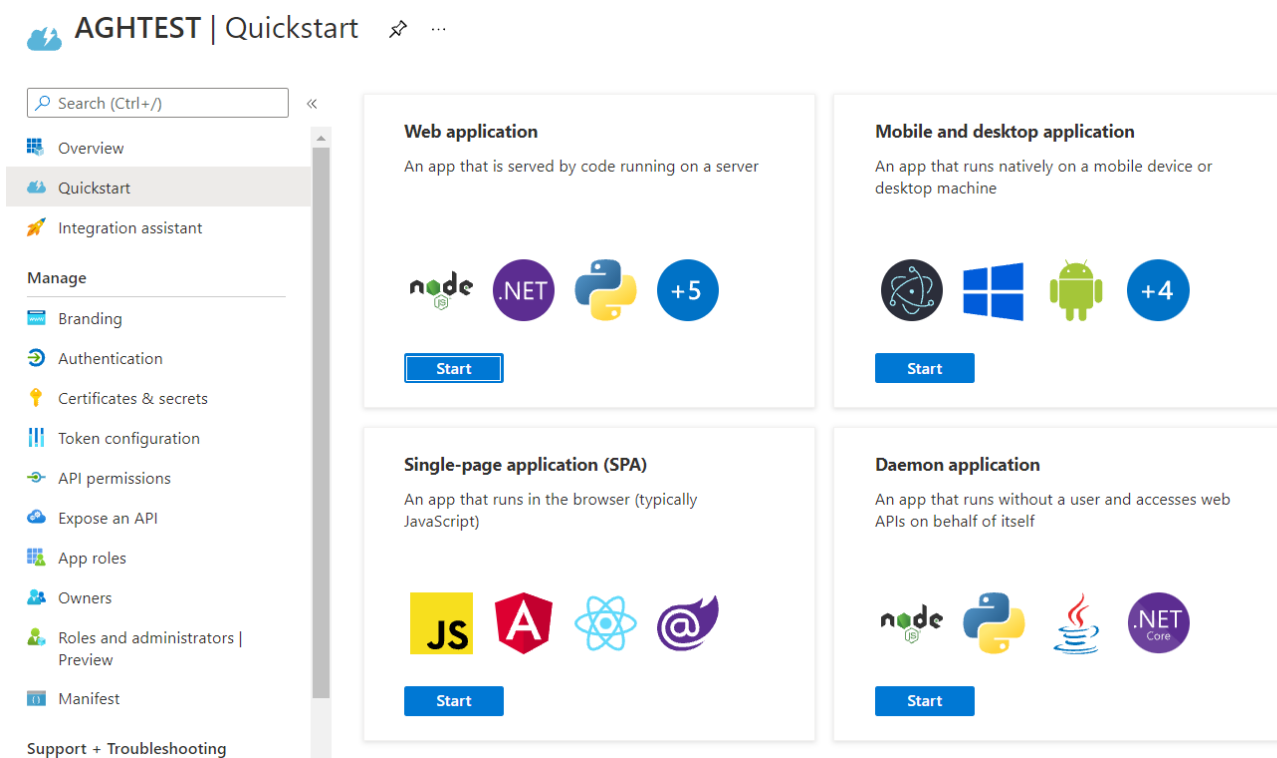
Supported account types : My organization only

Welcome to the new and improved App registrations. Looking to learn how it's changed from App registrations (Legacy)? [Learn more](#)

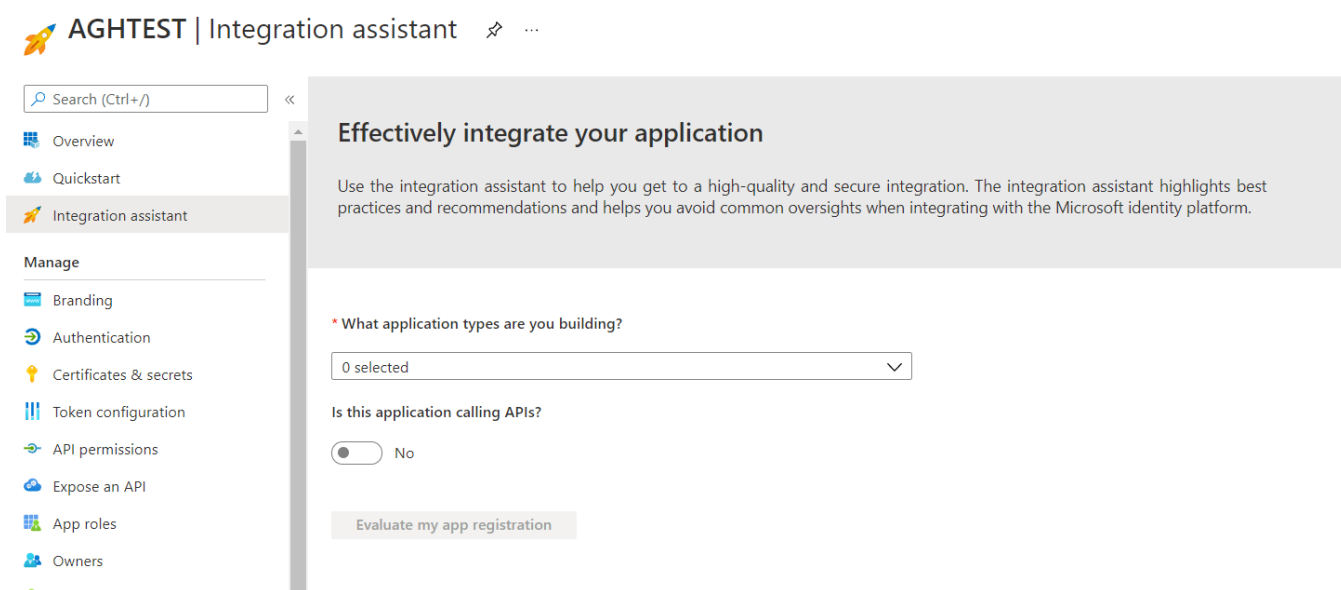
Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure AD Graph. We will continue to provide technical support and security updates but we will no longer provide feature updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph. [Learn more](#)

[Get Started](#) [Documentation](#)

6. Zaraz po zarejestrowaniu aplikacji zobaczymy jak wygląda interfejs użytkownika. Teraz, zanim zagłębimy się w szczegóły pisania aplikacji która zabezpieczymy, zrozummy, co oznaczają te wszystkie rzeczy po lewej stronie, jakie są różne rzeczy dostępne podczas rejestracji aplikacji.



7. **Szybki start** to dla świetny sposób na bardzo szybkie rozpoczęcie pisania aplikacji. Wrócimy tu za chwilę.



8. **Asystent integracji** pozwala postępować zgodnie z najlepszymi praktykami. Odpowiadamy więc tutaj na kilka prostych pytań, a następnie podpowiadane nam są jakie są najlepsze praktyki, czy je stosujemy, czy nie.

Search (Ctrl+/) << Save Discard Got feedback?

- Overview
- Quickstart
- Integration assistant
- Manage
 - Branding
 - Authentication
 - Certificates & secrets
 - Token configuration
 - API permissions
 - Expose an API
 - App roles

Name * ⓘ AGHTEST

Logo None provided

Upload new logo ⓘ Select a file

Home page URL ⓘ e.g. https://example.com

Terms of service URL ⓘ e.g. https://example.com/termsofservice

Privacy statement URL ⓘ e.g. https://example.com/privacystatement

Publisher domain ⓘ janekbaumgart@gmail.onmicrosoft.com [Update domain](#)

The application's consent screen will show 'Unverified'.
[Learn more about publisher domain](#)

Publisher verification

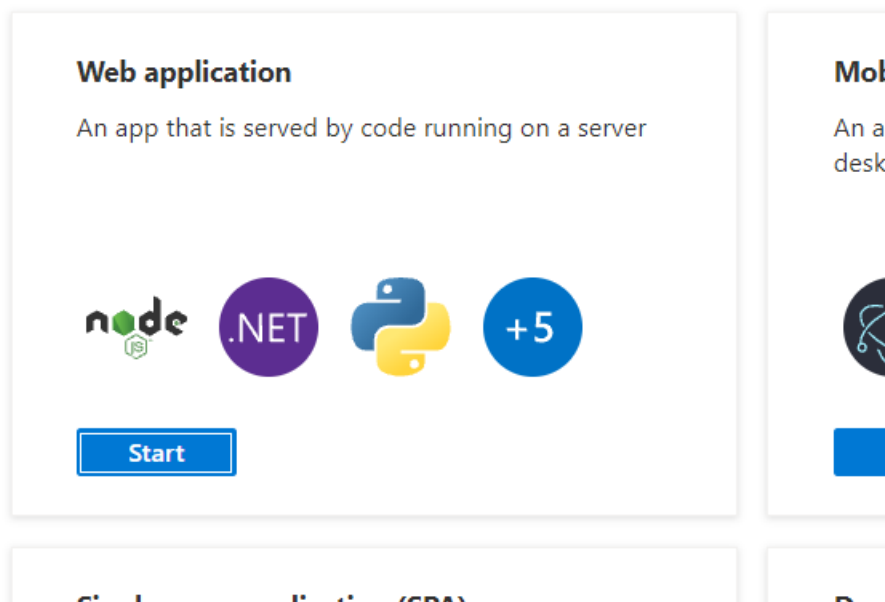
- 9.
10. **Branding**, jak sama nazwa wskazuje, to branding aplikacji, adres URL strony głównej logo itp. Nasza aplikacja będzie traktowana jako niezwyfikowana w tym miejscu. Gdy ktoś spróbuje zainstalować naszą aplikację, tak jakby była to aplikacja dostępna dla wszystkich z kontami Microsoft, po prostu pokaże się komunikat, że aplikacja jest niezwyfikowana. Użytkownicy nadal mogą z niej korzystać, jeśli duża firma pisze kilka aplikacji dla Microsoft Entra ID i chce, aby ich marka była z nią powiązana, aby ludzie mogli jej ufać, to zwyfikują się jako wydawca.
11. **Uwierzytelnianie** – tutaj dodajemy różne rodzaje uwierzytelniania.
12. **Certyfikaty i sekrety** to miejsce, w którym mogę zarządzać certyfikatami i sekretami, jak sama nazwa wskazuje, i będę tego potrzebować do aplikacji, takich jak aplikacja internetowa.
13. **Konfiguracja tokenów** – na początku wydawane tokeny mają określony format i możemy je w tym miejscu dowolnie zmieniać.
14. **Uprawnienia API** – Tutaj konfigurujemy jakie API może wywoływać moja aplikacja. Mamy dwie główne możliwości, uprawnienia delegowane to te, w których ważna jest tożsamość użytkownika, a uprawnienia aplikacji to te, w których tożsamość użytkownika nie jest ważna.
15. **Opublikuj API** - to miejsce, w którym możemy zdecydować żeby wystawić API.
16. **Właściciel** - Domyślnie właścicielem jest osoba, która utworzyła aplikację, ale możemy dodać więcej właścicieli, aby mogli zarządzać aplikacją za nas.
17. **Role i administratorzy** - umożliwia zarządzanie rolami i administratorami dla tej aplikacji.
18. **Manifest** jest reprezentacją JSON konfiguracji naszej aplikacji i udostępnia dalsze funkcje, które nie są widoczne w interfejsie użytkownika

2. Konfiguracja uwierzytelniania

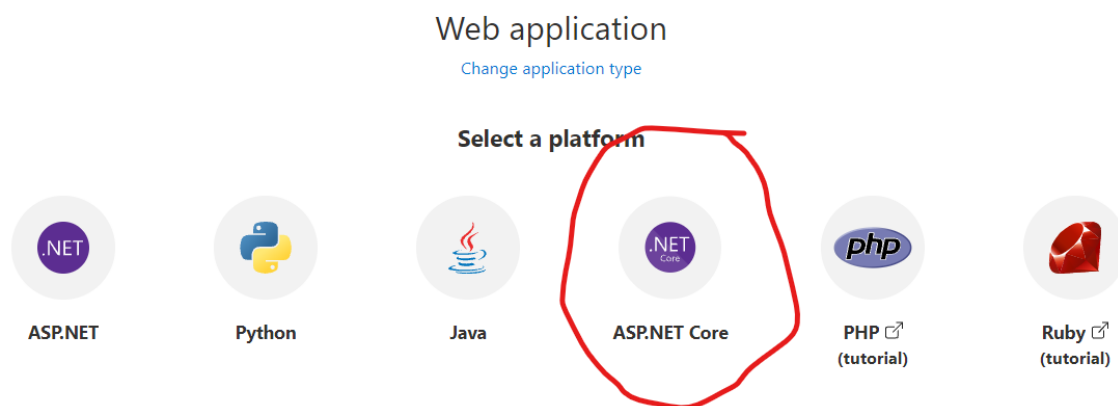
1. W poprzednim kroku zarejestrowaliśmy aplikację w Microsoft Entra ID.

2. Uruchomimy teraz prostą aplikację, najłatwiej to zrobić, przechodząc do zakładki Szybki start.
3. Tutaj zostaniesz poproszony o wybranie rodzaju aplikacji, którą chcemy napisać. Powiedzmy, że napiszemy aplikację internetową.

What type of application are you building?

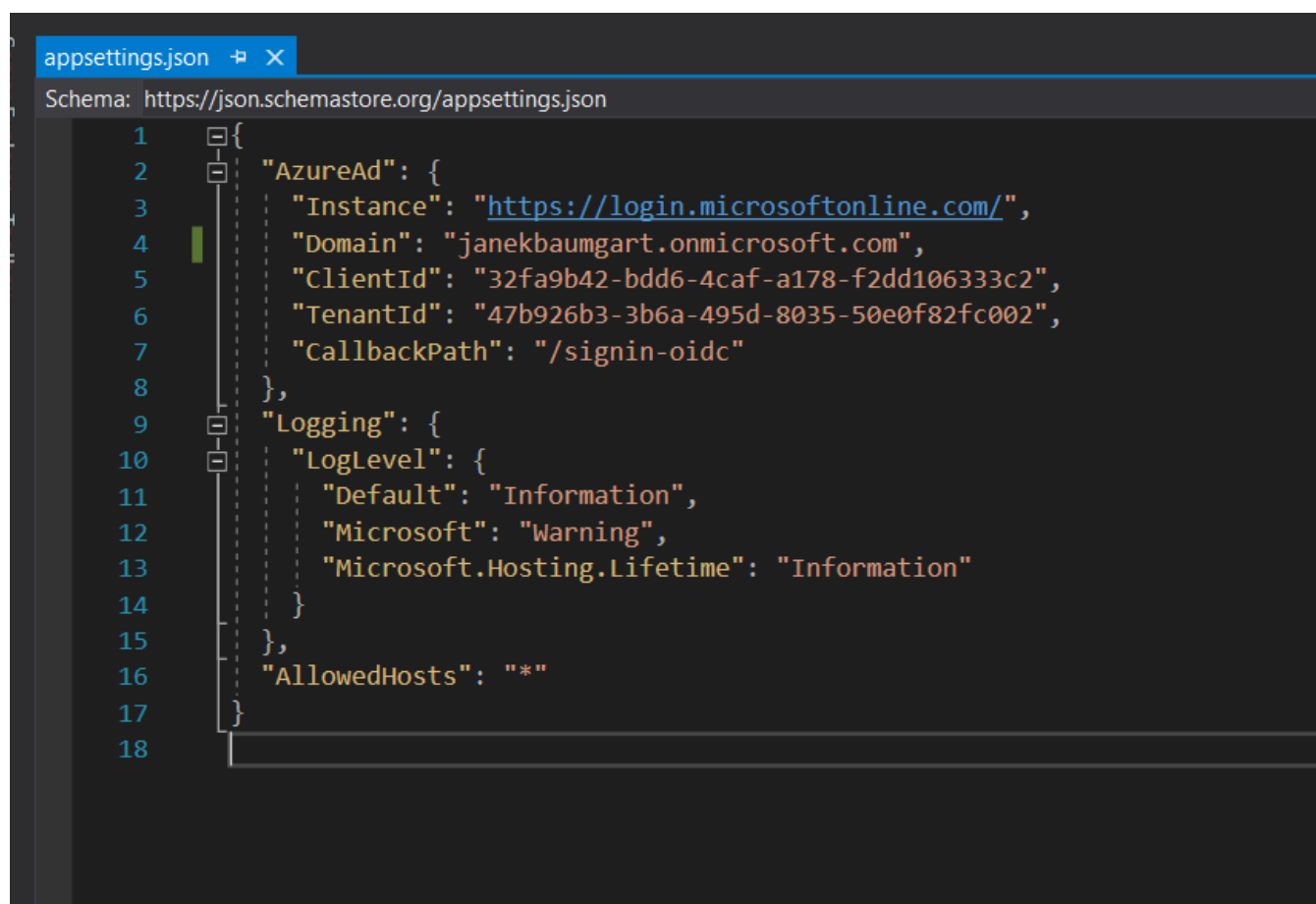


4. Poniżej pokazano kilka platform, wybierzemy ASP.NET Core.



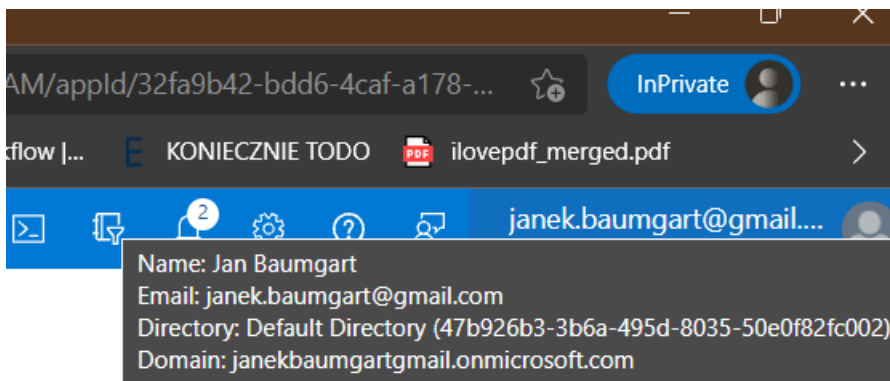
5. Pokaże się przewodnik, który przeprowadzi nas przez proces konfigurowania uwierzytelniania dla naszej aplikacji, o ile jest napisana w ASP.NET Core. Ale oczywiście możesz również wybrać inną platformę zależnie od technologii w której pracujecie.
6. Przewodnik jest interaktywny, więc można go przeczytać i postępować zgodnie z instrukcjami lub uruchamiać komendy bezpośrednio z przewodnika.

7. Kliknijmy „Wykonaj te zmiany dla mnie” („Make this change for me”) i zobaczymy dokładnie jakie zmiany dokona przewodnik. A więc to, co się tutaj dzieje:
 - a. Azure oczekuje, że nasza aplikacja będzie działać na <https://localhost:44321> i że będzie nasłuchiwać usługi Azure AD. Dlatego gdy usługa Azure AD uwierzyteli użytkownika, wyśle nasz token do tego adresu URL logowania.
 - b. Wylogowanie następuje pod podanym adresem URL
 - c. Oraz że prosimy o token identyfikacyjny.
8. Wracamy do przewodnika, i pobieramy wstępnie przygotowaną próbkę kodu. Tak więc ten przykładowy kod jest zasadniczo skonfigurowany do uruchamiania przy użyciu platformy .NET Core.
9. Otwórzmy folder i rozpakujmy jego zawartość.
10. Tutaj znajdujemy rozwiązanie Visual Studio. Gdy otworzymy to rozwiązanie Visual Studio, zobaczymy, że jest w nim jeden projekt .NET Core.
11. Nasz wymaga zależności od niektórych pakietów NuGet. Gdy otworzyć Dependencies, przejść do Packages, zobaczymy, że używamy zależności Microsoft.Identity.Web i Microsoft.Identity.Web.UI.



```
1  {
2    "AzureAd": {
3      "Instance": "https://login.microsoftonline.com/",
4      "Domain": "janekbaumgart.onmicrosoft.com",
5      "ClientId": "32fa9b42-bdd6-4caf-a178-f2dd106333c2",
6      "TenantId": "47b926b3-3b6a-495d-8035-50e0f82fc002",
7      "CallbackPath": "/signin-oidc"
8    },
9    "Logging": {
10     "LogLevel": {
11       "Default": "Information",
12       "Microsoft": "Warning",
13       "Microsoft.Hosting.Lifetime": "Information"
14     }
15   },
16   "AllowedHosts": "*"
17 }
18
```

12. Zjrzyjmy dalej do appsettings.json. A w appsettings.json niektóre ustawienia zostały już dla nas skonfigurowane, ClientId i TenantId.
13. Przejdźmy do naszej aplikacji w Azure AD i zobaczmy, skąd pochodzą.
14. Przejdziemy do karty Przegląd, a ClientId i TenantId to są właśnie te wartości.



15. Nadal musimy skonfigurować domenę. Moja domena to janekbaumgart.onmicrosoft.com. Skąd otrzymuję tę wartość? Najedź na swój mail w prawym górnym rogu panelu i zobaczysz to w podpowiedzi.
16. A teraz po prostu naciśniemy przycisk Uruchom, aby uruchomić tę aplikację.
17. Zostaniemy poproszeni o zaufanie certyfikatu SSL.
18. I już uruchomi się nasz serwer WWW i nasza aplikacja. W przeglądarce zostaniemy przekierowani do localhost:44321.
19. Nasze uwierzytelnianie już działa!

3. Konfiguracja Autoryzacji za pomocą grup.

1. Aby skonfigurować autoryzację dla naszej aplikacji która potrafi się już uwierzytelniać. Musimy zacząć od konfiguracji tokenów w Microsoft Entra ID. Aby to zrobić przejdźmy do App Registrations i wybierzmy naszą aplikację z listy.
2. Następnie wybieramy opcję Konfiguracja Tokenów (Token Configuration) i klikamy Dodaj konfigurację grupy (Add group claim).

AGHTEST | Token configuration

[Got feedback?](#)

Overview

Quickstart

Integration assistant

Manage

Branding

Authentication

Certificates & secrets

Token configuration

API permissions

Expose an API

App roles

Owners

Roles and administrators | Preview

Manifest

Optional claims

Optional claims are used to configure additional information which is returned in one or more tokens.

[+ Add optional claim](#) [+ Add groups claim](#)


Claim ↑↓	Description
No results.	

3.

4. Następnie wybieramy opcję Grupy Bezpieczeństwa (Security Groups)

Edit groups claim



 Adding the groups claim applies to Access, ID, and SAML token types. [Learn more](#)

Select group types to include in Access, ID, and SAML tokens.

- ☒ Security groups
- ☐ Directory roles
- ☐ All groups (includes distribution lists but not groups assigned to the application)
- ☐ Groups assigned to the application

Customize token properties by type

▼ ID

▼ Access

▼ SAML

- 5.
6. Pozostałą konfigurację pozostawiamy bez zmian.
7. Musimy teraz utworzyć grupę która będzie zarządzała naszymi dostępami. Aby to zrobić przechodzimy do Microsoft Entra ID i zakładka Grupy (Groups).
8. Następnie klikamy Dodaj grupę (New group)

[Home](#) > [Default Directory](#) >

Groups | All groups

Default Directory - Azure Active Directory




 New group



Download groups




Delete

 All groups

 Deleted groups



This page includes previews available for your evaluation

9.  Diagnose and solve problems

10. Należy podać nazwę grupy i kliknąć Utwórz (Create)
11. Teraz przechodzimy do naszego kodu stworzonej uprzednio aplikacji. I dodajemy następujący kod do pliku Startup.cs

12.

```
29 // This method gets called by the runtime. Use this method to add services to the container.
30 public void ConfigureServices(IServiceCollection services)
31 {
32     services.AddAuthentication(OpenIdConnectDefaults.AuthenticationScheme)
33         .AddMicrosoftIdentityWebApp(Configuration.GetSection("AzureAd"));
34
35     services.AddAuthorization(options => {
36         options.AddPolicy("Admin-Only", p => {
37             p.RequireClaim("groups", "88e8f5b6-3830-4289-ade8-3ef4630accac");
38         });
39     });
40
41     services.AddControllersWithViews(options =>
42     {
43         var policy = new AuthorizationPolicyBuilder()
44             .RequireAuthenticatedUser()
45             .Build();
```

13. Dzięki temu dodaliśmy nową polisę autoryzacji nazwaną Admin-Only której wymogiem jest posiadanie odpowiedniej grupy o podanym ID.
14. Następnie przechodzimy do pliku HomeController.ps (w folderze Controllers) i edytujemy następująco:

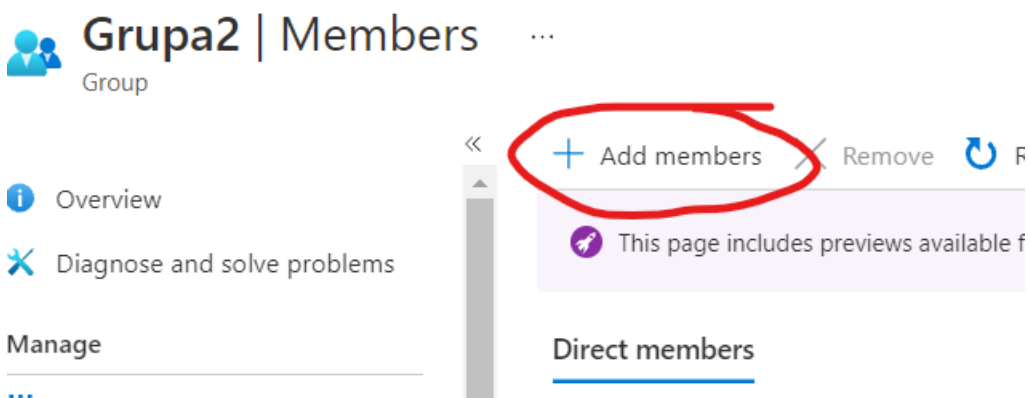
```
19 {
20     _logger = logger;
21 }
22
23 public IActionResult Index()
24 {
25     return View();
26 }
27
28 [Authorize("Admin-Only")]
29 public IActionResult Privacy()
30 {
31     return View();
32 }
33
34 [AllowAnonymous]
35 [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]
36 public IActionResult Error()
37 {
```

15. Możemy uruchomić program, należy zwrócić uwagę że nie uzyskamy dostępu do części naszej aplikacji o nazwie Privacy, ponieważ nie jesteśmy członkami utworzonej grupy.

Access denied

You do not have access to this resource.

16. Aby uzyskać tam dostęp musimy teraz dodać swojego użytkownika do naszej grupy. Aby to zrobić przechodzimy do Microsoft Entra ID, następnie zakładka Groups i wybieramy z listy swoją grupę.
17. Następnie z lewej strony wybieramy zakładkę Członkowie (Members) i dodajemy siebie



18. Następnie możemy wrócić do naszego kodu i uruchomić go ponownie. Tym razem już ukaże nam się prawidłowa strona.

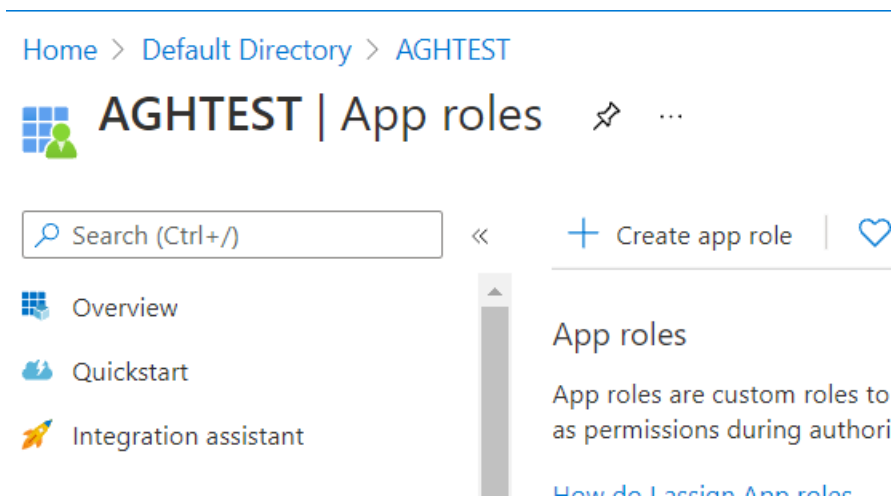
Privacy Policy

Use this page to detail your site's privacy policy.

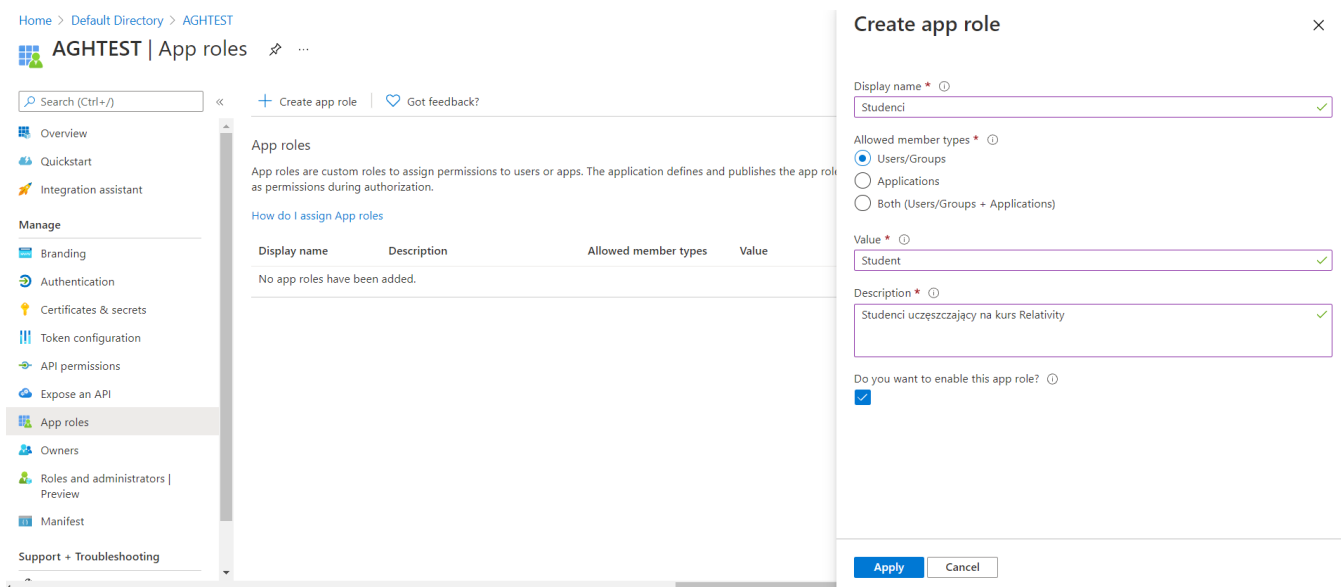
4. Konfiguracja autoryzacji za pomocą roli

Proces konfiguracji autoryzacji za pomocą roli zaczynamy od utworzenia roli. W tym celu należy:

19. Zalogować się do platformy Azure -> wybrać Microsoft Entra ID -> Rejestrację Aplikacji (App Registration)
20. Następnie z listy wybrać naszą aplikację.
21. Po wybraniu naszej aplikacji należy wybrać z lewego panelu opcję „Role Aplikacji” (App Roles)
22. Następnie kliknąć Stwórz Rolę w Aplikacji

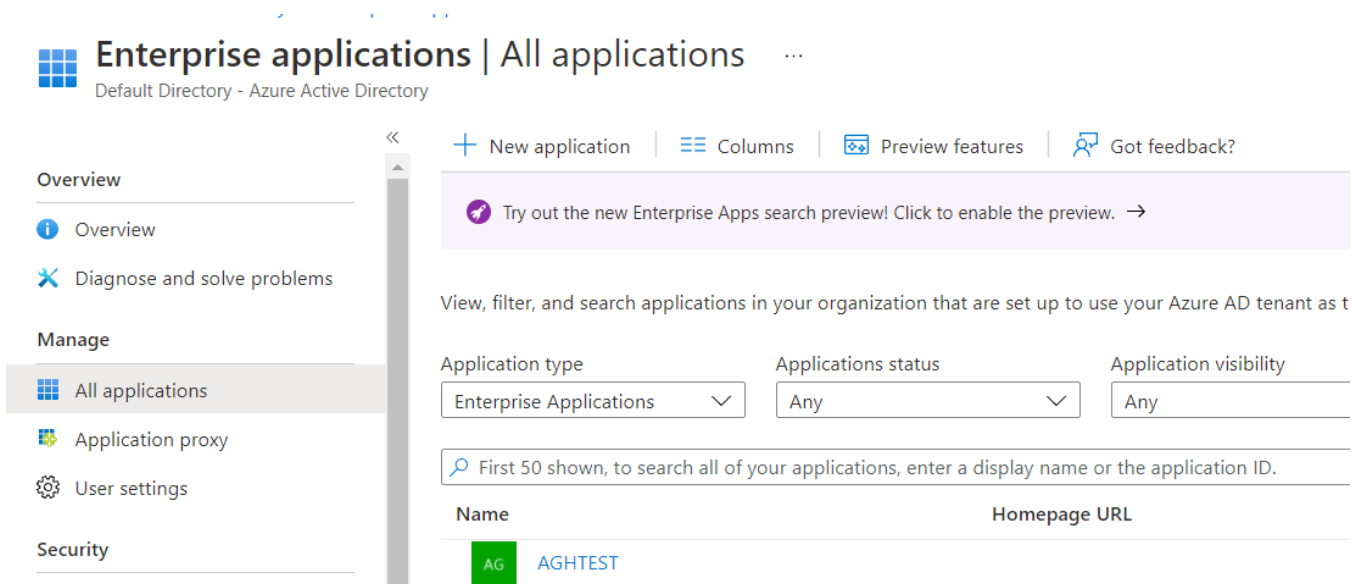


23. Następnie uzupełnić odpowiednio dane
 - a. Nazwa roli: Studenci
 - b. Wybrać Użytkownicy/Grupy (Users/Groups)
 - c. Wartość: Student
 - d. Opis: Grupa dla studentów uczęszczających na Kurs Relativity
 - e. i kliknąć Zatwierdź.



24. Następnie należy przejść do Microsoft Entra ID i tym razem z prawej strony wybrać „Aplikacje przedsiębiorstwa” (Enterprise Applications). **NIE Rejestracje aplikacji (App registrations)**

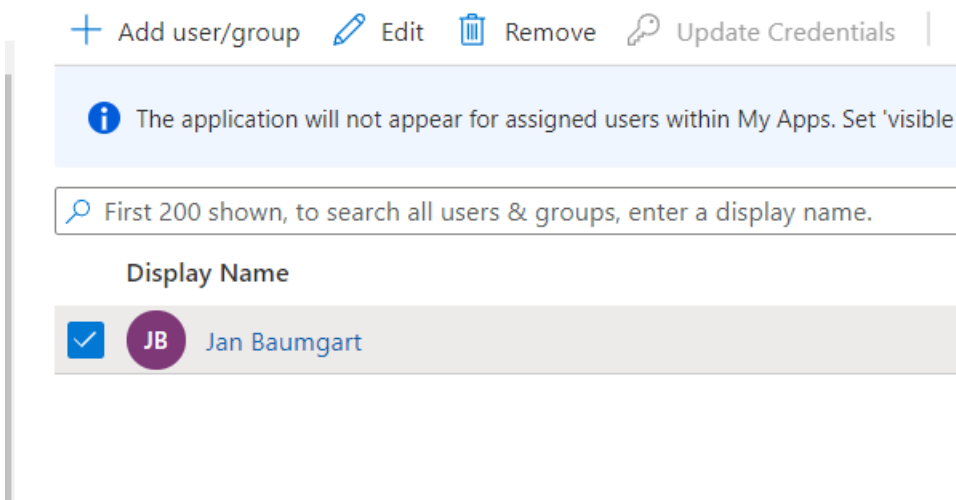
25. Następnie wybieramy naszą aplikację z listy



26. Kolejny krokiem jest wybranie z lewej strony zakładki Użytkownicy i Grupy

27. Zaznaczamy widocznego użytkownika i klikamy Usuń (Remove)

in groups



28. Następnie klikamy Dodaj użytkownika/grupę (Add user/group)

29. I dodajemy swojego użytkownika ponownie tym razem już z odpowiednią Rolą.

[Home](#) > [Default Directory](#) > [Enterprise applications](#) > [AGHTEST](#) >

Add Assignment

Default Directory

⚠ Groups are not available for assignment due to your Active Directory plan level. You can assign individual users to the application.

Users

1 user selected.

Select a role

Student

30. Następnie wracamy do Visual Studio i dodajemy następujący kod:

```
{
    _logger = logger;
}

[Authorize(Roles = "Student")]
public IActionResult Index()
{
    return View();
}
```

```

35
36     services.AddAuthorization(options => {
37         options.AddPolicy("Admin-Only", p => {
38             p.RequireClaim("groups", "6e2196f7-8179-408f-90c3-d1db2214daa4");
39         });
40
41         options.AddPolicy("Student", p => {
42             p.RequireRole("Student");
43         });
44     });
45

```

31. Jak widzimy naszą Rolą nie jest nazwa Roli nadana w Portalu Azure, lecz jej wartość. (**Rola nazwana była Studenti, lecz jej wartość była Student**)

32. Po uruchomieniu kodu widzimy że mamy dostęp do strony głównej, widzimy że Rola Studenti zadziała prawidłowo.

WebApp_OpenIDConnect_Dot
Net

Home Privacy Hello
janek.baumgart@gmail.com! Sign
out

Welcome

Learn about [building Web apps with ASP.NET Core](#).

W ten sposób konfigurujemy autoryzację za pomocą roli i grup.