# Auth0 in ASP.NET Core MVC applicatie:

# Role based authentication

## Student:

Hellemans Nick

## Probleemstelling

In het project moesten we gebruik maken van Auth0 om de authenticatie te voorzien voor de webapplicaties van de werknemers en admins. Het admin dashboard, dat een ASP.NET Core MVC applicatie is, moet afgeschermd worden voor onbevoegde werknemers zodat enkel de werknemers met de correcte bevoegdheden, de admins, toegang hebben tot de webapplicatie. Dit hebben we geïmplementeerd aan de hand van “Role based authenticatie”. Elke werknemer krijgt een specifieke rol toegekend dat bepaald welke rechten hij/zij heeft en tot welke resource hij/zij toegang tot heeft.

De documentatie dat Auth0 voorziet om authenticatie, en al zeker authenticatie aan de hand van rollen, te verwerken in je project, is helemaal niet volledig en correct om het tot een goed einde te kunnen brengen. Hierdoor moest ik op zoek naar andere tutorials en deze waren niet altijd makkelijk te vinden. Uiteindelijk door veel trial & error en het doorlopen van meerdere tutorials/forums heb ik het werkende gekregen. In dit stappenplan zal ik alle nodige stappen en code vermelden die nodig zijn om in je ASP.NET Core MVC applicatie authenticatie te voorzien op basis van zelf gespecifieerde rollen.

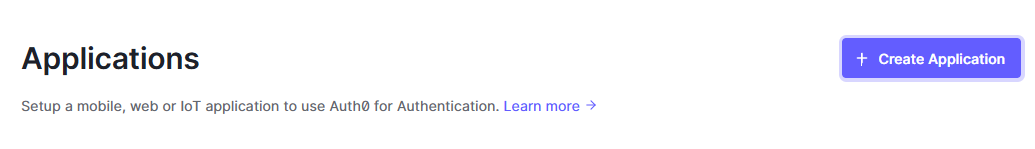
## Stappenplan

**Basic Setup**

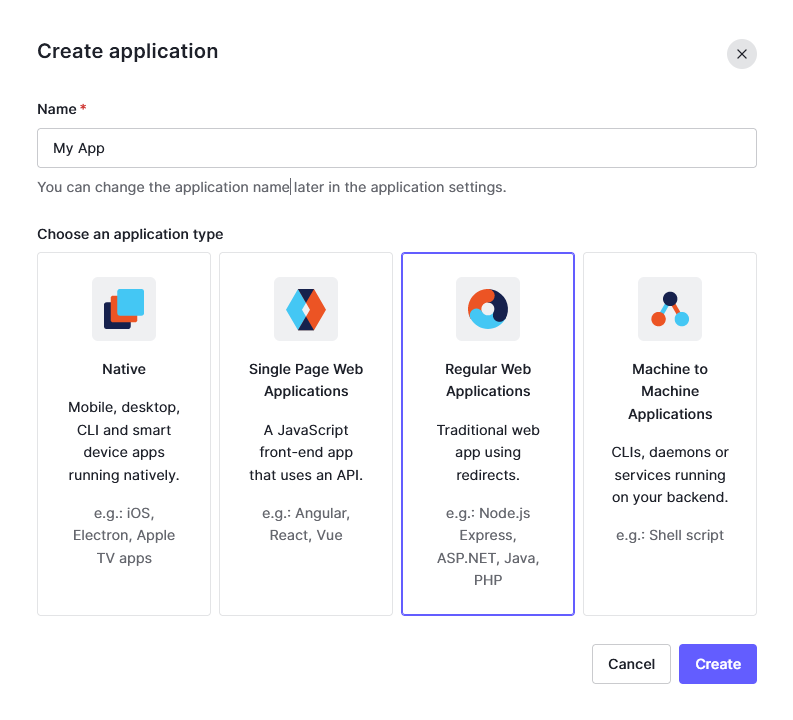
Een eerste belangrijke stap is het aanmaken van een Auth0 account, als je deze nog niet zou hebben, kan je dit doen op onderstaande link:

<https://auth0.com/signup>

Eenmaal dit gebeurd is, moet je in het dashboard navigeren naar de applicaties pagina. Hier kan je een nieuwe applicatie toevoegen.



Maak een nieuw applicatie aan met een naam naar keuze en kies “Regular Web Applications” in het keuzemenu aangezien we werken met een ASP.NET applicatie.



Nu zullen we enkele settings aanpassen in de juist aangemaakt applicatie. Ga naar de sectie “Application URIs” en in het veldje voor “Allowed Callback URLs” voeg je de URL waar van je MVC-applicatie toe met een “/callback” toegevoegd achteraan de URL bv:

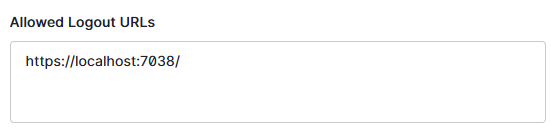
<https://localhost:7038/callback>

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Dit zorgt ervoor dat Auth0 weet naar welke URLs het mag terugkeren nadat de authenticatie volledig doorlopen is.

In dezelfde sectie voeg je bij “Allowed Logout URLs” de URL toe waarnaar je je gebruikers wilt doorverwijzen na uitloggen of welke URLs geldig zijn om naar te redirecten naar uitloggen, bv:



Vergeet zeker en vast niet de veranderingen op te slaan!

Nu is het tijd om onze Auth0 applicatie te connecteren met de feitelijke MVC applicatie. In je “appsettings.json” configuratie bestand voeg je volgende zaken toe:

* Domain
* ClientId

Deze zijn beide te vinden in het Auth0 dashboard bij je applicatie.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Vervang de placeholders met de juiste waarden vanuit je Auth0 dashboard. De basic setup is nu klaar en kunnen we beginnen met het effectief koppelen van de MVC-applicatie met Auth0.

**Connecteer Auth0 en MVC project**

Installeer de Auth0 SDK in je MVC-project:



In het “program.cs” bestand voeg je onderstaande code toe:





Eens dit in orde is, kunnen we de endpoints voorzien voor in – en uitloggen. Dit doen we het liefste in een aparte controller, bv. “AccountController”.





Het is natuurlijk vanzelfspreken dat de gebruikers deze endpoints moeten kunnen aanroepen vanuit de webapplicatie dit kan je doen door een in – en uitlogknop te voorzien dat zich aanpast naargeland de gebruiker ingelogd is of niet. Hieronder vind je wat voorbeeldcode hoe je dit zou kunnen aanpakken in je views.



Met deze stappen hebben we nu authenticatie voorzien in onze MVC-applicatie. Het volgende deel gaat over hoe we onze endpoints kunnen beschermen op basis van rollen.

**Role based authenticatie**

Vooraleer we authenticatie kunnen toevoegen op basis van rollen is het natuurlijk belangrijk dat de nodige rollen zijn aangemaakt in het Auth0 dashboard en dat de gebruikers gekoppeld zijn aan de correcte rol. De documentatie van Auth0 is hier duidelijk genoeg hoe dit moet gebeuren mocht dit niet duidelijk zijn:

<https://auth0.com/docs/manage-users/access-control/configure-core-rbac/rbac-users/assign-roles-to-users>

Eens de rollen en gebruikers in orde zijn, moeten we in het Auth0 dashboard een nieuwe regel toevoegen (https://auth0.com/docs/customize/rules). Dit zorgt ervoor dat de rol van de gebruiker toegevoegd wordt aan zijn ID-token. Zo kan de backend deze gebruiken om autorisatie na te kijken. Dit is dus een zeer belangrijke stap, als je backend niet aan de rol kan van de gebruiker kan deze de rol ook niet gebruiken om te controleren of de gebruiker wel toegang heeft of niet tot een bepaalde resource. Om een nieuwe regel toe te voegen navigeer je naar:

**Auth Pipeline > Rules > Empty Rule**

Voeg volgende code toe aan je nieuwe regel:

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Hier nog eens dezelfde code om makkelijk te kunnen kopiëren (dubbelklik op de afbeelding):



Alles staat nu klaar en we kunnen gebruik maken van role based autorisatie. Dit doen we door op alle acties waarvoor we een specifieke rol willen controleren een attribuut toe te voegen zoals in onderstaand voorbeeld.

