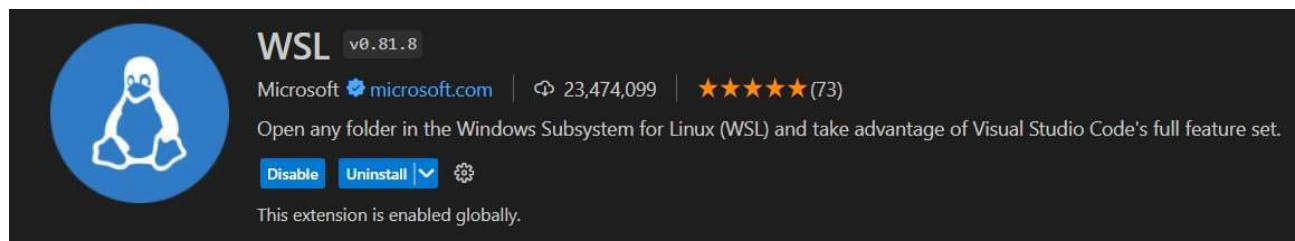
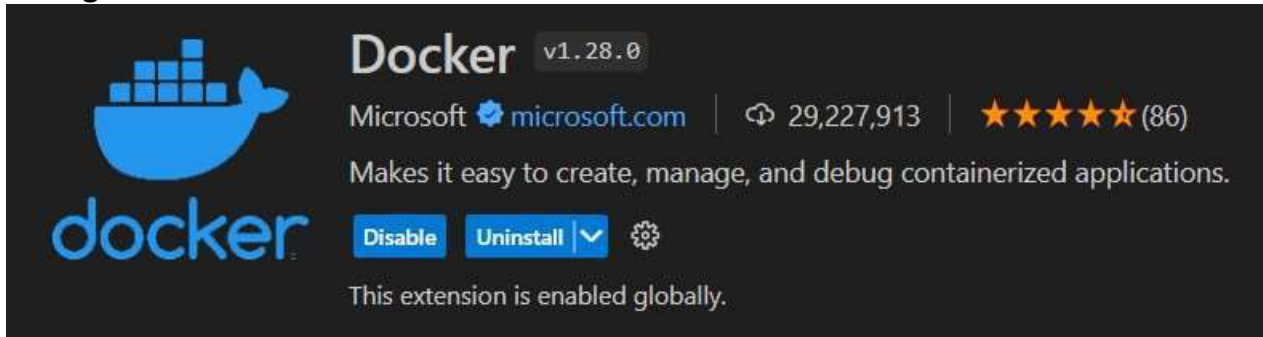


TalentenPortaal – Repository TalentenPortaalAPITest + PostgreSQL database

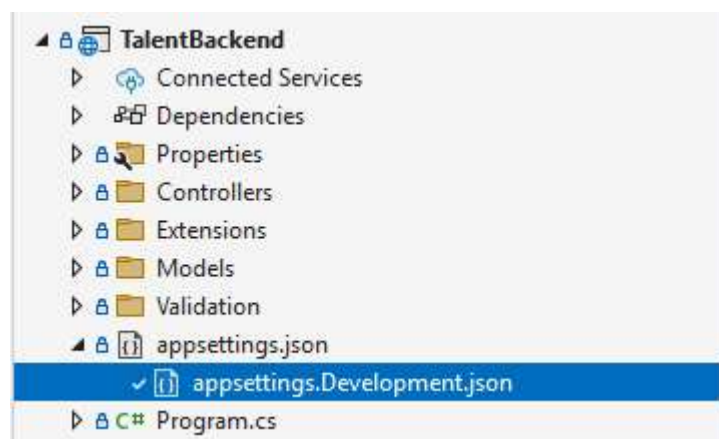
1. Download en installeer Docker Desktop. Start Docker Desktop na installatie

<https://www.docker.com/products/docker-desktop/>

2. Voeg in Visual Studio Code de 2 extensies toe:



3. Controleer de database connection-string van het project in Visual Studio 2022 – zie het bestand *appsettings.Development.json* van het project *TalentBackend*



Afbeelding hieronder: *appsettings.Development.json* – sectie *connectionStrings*

```
"ConnectionStrings": {  
  // localdb  
  "TalentPortal-Localdb": "Host=localhost;5432;Password=Testen2025!;Persist Security Info=True;Username=postgres;Database=TalentenPortaal"  
}
```

In dit voorbeeld is de database connection-string:

"TalentPortal-Localdb": "Host=localhost;5432;Password=Testen2025!;Persist Security Info=True;Username=postgres;Database=TalentenPortaal"

4. Maak een directory aan voor het docker-compose bestand. Bijvoorbeeld: PostgreSQL_talenten_portaal_test

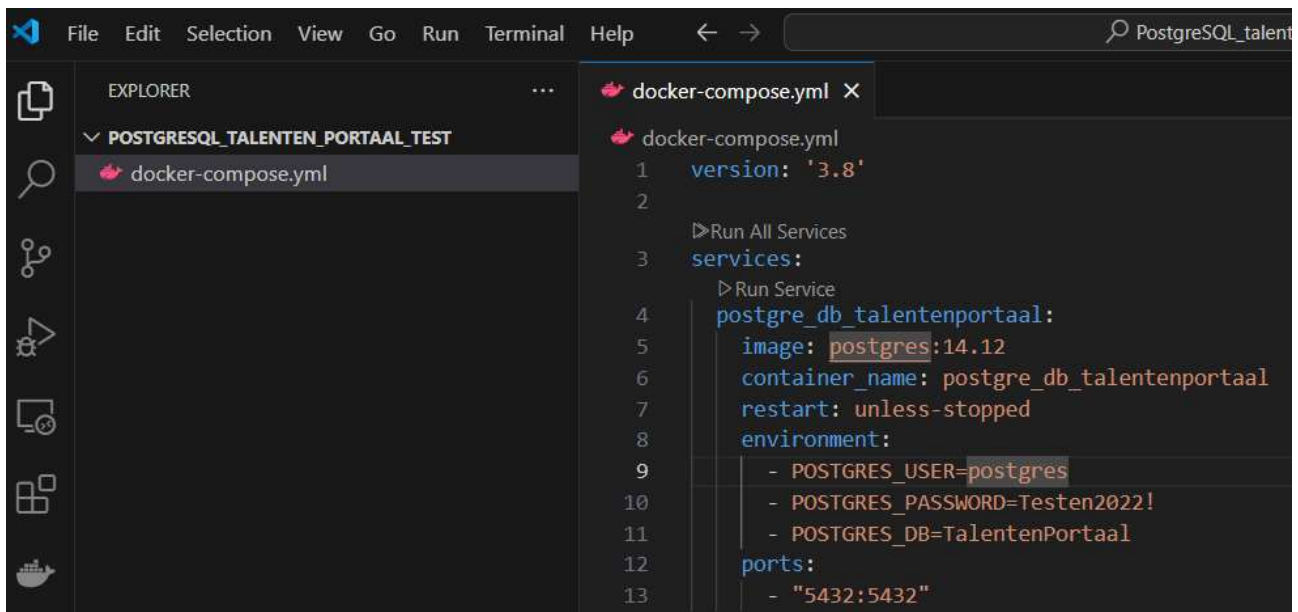
Maak in deze directory het bestand **docker-compose.yml** aan en plak hierin de inhoud hieronder.

Zorg ervoor dat **POSTGRES_USER**, **POSTGRES_PASSWORD** en **POSTGRES_DB** overeenkomen met de gegevens in de connection-string hierboven.

```
version: '3.8'

services:
  postgres_db_talentenportaal:
    image: postgres:14.12
    container_name: postgres_db_talentenportaal
    restart: unless-stopped
    environment:
      - POSTGRES_USER=postgres
      - POSTGRES_PASSWORD=Testen2025!
      - POSTGRES_DB=TalentenPortaal
    ports:
      - "5432:5432"
```

Visual Studio Code ziet er nu zo uit:



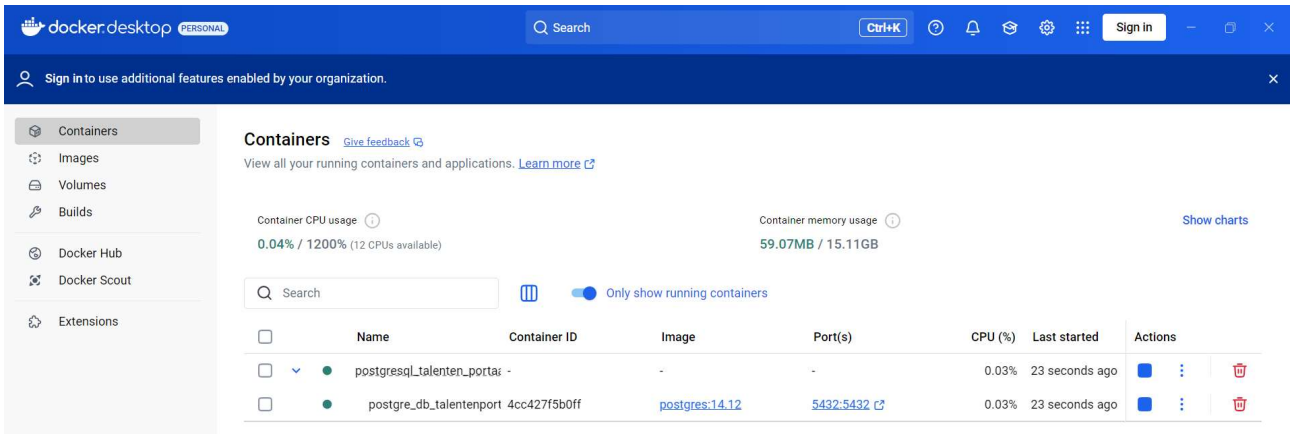
4. Open de terminal in VS code via het menu: Terminal → New Terminal

5. Type het volgende commando in de nieuwe terminal en druk op Enter:

docker-compose up --build -d



6. Wanneer dit commando klaar is, dan is de Docker-container in Docker Desktop te zien:

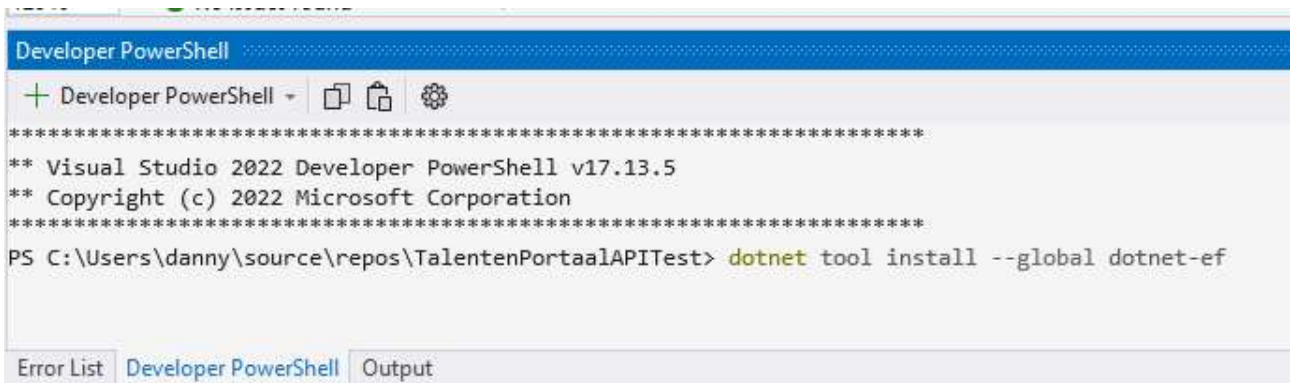


7. Ga naar Visual Studio 2022 en installeer de dotnet EF tool met het commando in de Developer Powershell:

dotnet tool install --global dotnet-ef

Of wanneer er een specifieke versie nodig is, bijvoorbeeld 8.0.4 voor dit project:

dotnet tool install --global dotnet-ef --version 8.0.4



Meer informatie:

<https://learn.microsoft.com/en-us/ef/core/cli/dotnet>

8. Controleer in het console of de installatie geslaagd is met het commando:

dotnet ef

9. De volgende stap is om de bestaande migratie in de database te plaatsen.

Het formaat van het commando is als volgt:

```
dotnet ef database update -p "PAD_VAN_JOUW_C_SHARP_WERKDIRECTORY\
TalentenPortaalAPITest\TalentDataAccess\TalentDataAccess.csproj" --startup-project
"PAD_VAN_JOUW_C_SHARP_WERKDIRECTORY\TalentenPortaalAPITest\TalentBackend\
TalentBackend.csproj"
```

Vervang in het commando hierboven **PAD_VAN_JOUW_C_SHARP_WERKDIRECTORY** met de werkdirectory op jouw computer.

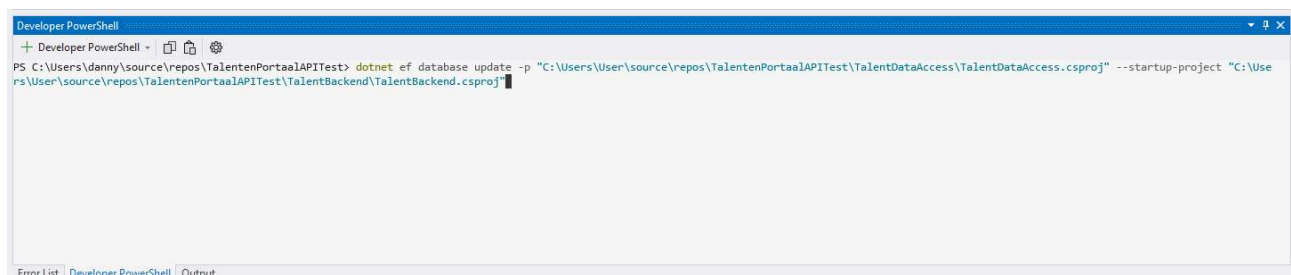
Voorbeeld: Wanneer het pad van de werkdirectory is:

C:\Users\someone\source\repos

Dan wordt het commando:

```
dotnet ef database update -p "C:\Users\someone\source\repos\TalentenPortaalAPITest\
TalentDataAccess\TalentDataAccess.csproj" --startup-project "C:\Users\someone\source\
repos\TalentenPortaalAPITest\TalentBackend\TalentBackend.csproj"
```

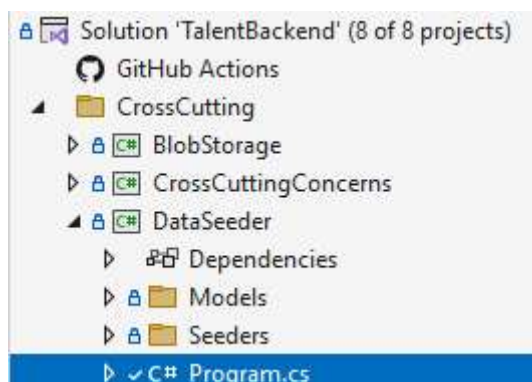
10. Kopieer en plak het aangepaste commando hierboven in de Developer Powershell van Visual Studio 2022 en druk op enter.



In het console staat na het uitvoeren van dit commando wat er tijdens de migratie is gebeurd.

11. De data voor dit project moet nog compleet gemaakt worden met de *DataSeeder*.

Controleer in *Program.cs* van de *DataSeeder* of de database connection-string dezelfde is als in *appsettings.Development.json*

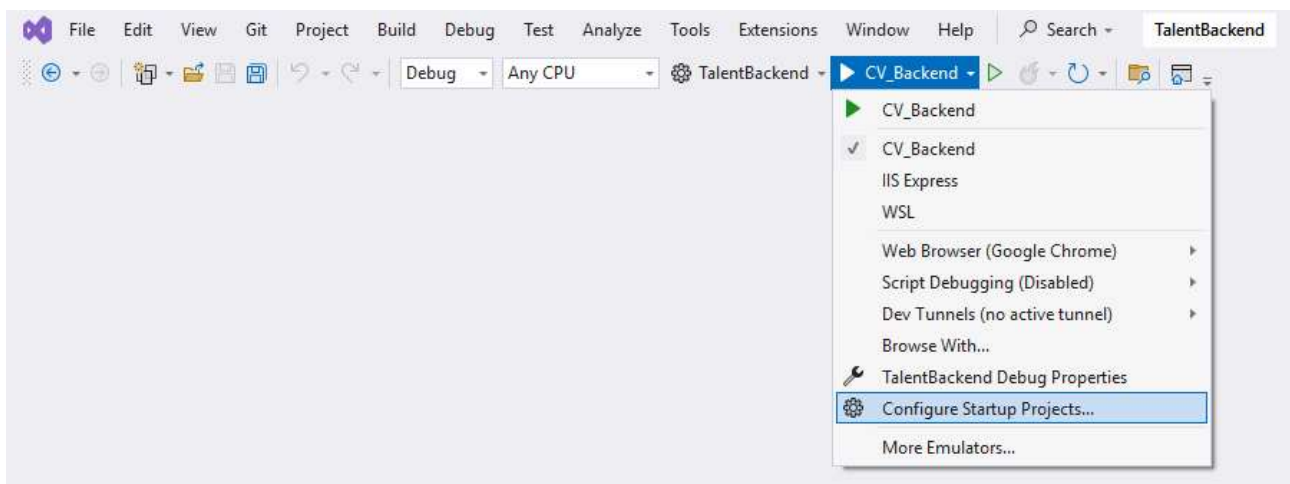


De connection-string is rond regel 15-16 te vinden:

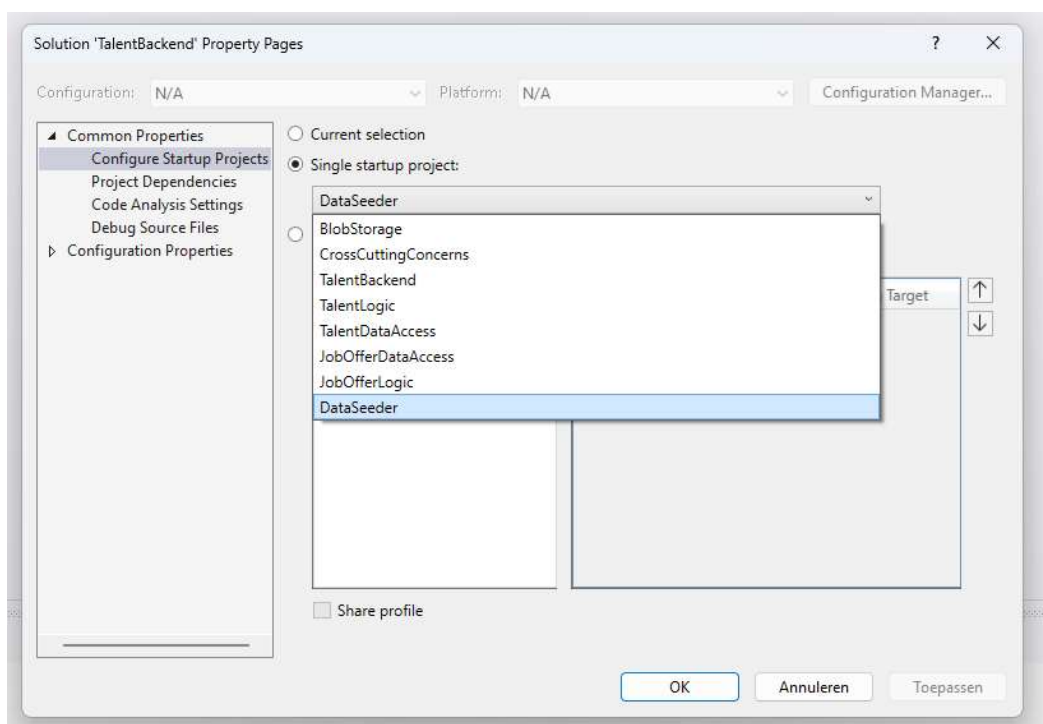
```
1 using DataSeeder.Seeders;
2 using Microsoft.EntityFrameworkCore;
3 using Microsoft.Extensions.DependencyInjection;
4 using System;
5 using System.Threading.Tasks;
6 using TalentDataAccess.DataAccess;
7
8 namespace DataSeeder
9 {
10     internal class Program
11     {
12         private static async Task Main(string[] args)
13         {
14             #if DEBUG
15                 var DatabaseConnectionString = "Host=localhost,5432;Password=Testen2025!;Persist Security Info=True;Username=postgres;Database=TalentenPortaal";
16             #else
17                 var DatabaseConnectionString = "Host=162.55.181.242;Port=5433;Password=dab2b61743ca034eae37;Persist Security Info=True;Username=postgres;Database=it-talenten-portaal-test";
18             #endif
19         }
20     }
21 }
```

Zorg ervoor dat de connection-string dezelfde is als in *appsettings.Development.json*

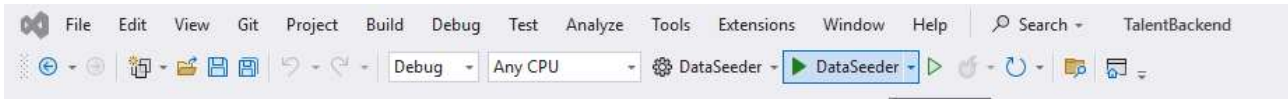
12. DataSeeder uitvoeren – klik in Visual Studio 2022 op de kleine triangel naast de startknop en ga naar *Configure Startup Projects*



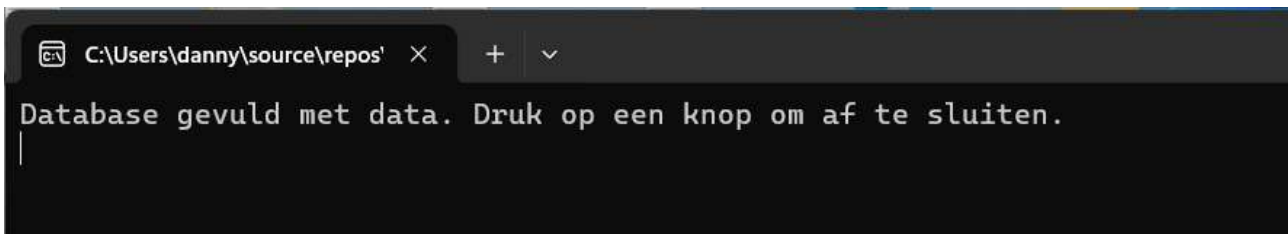
13. Selecteer in het volgende menu *DataSeeder* en klik op *OK*.



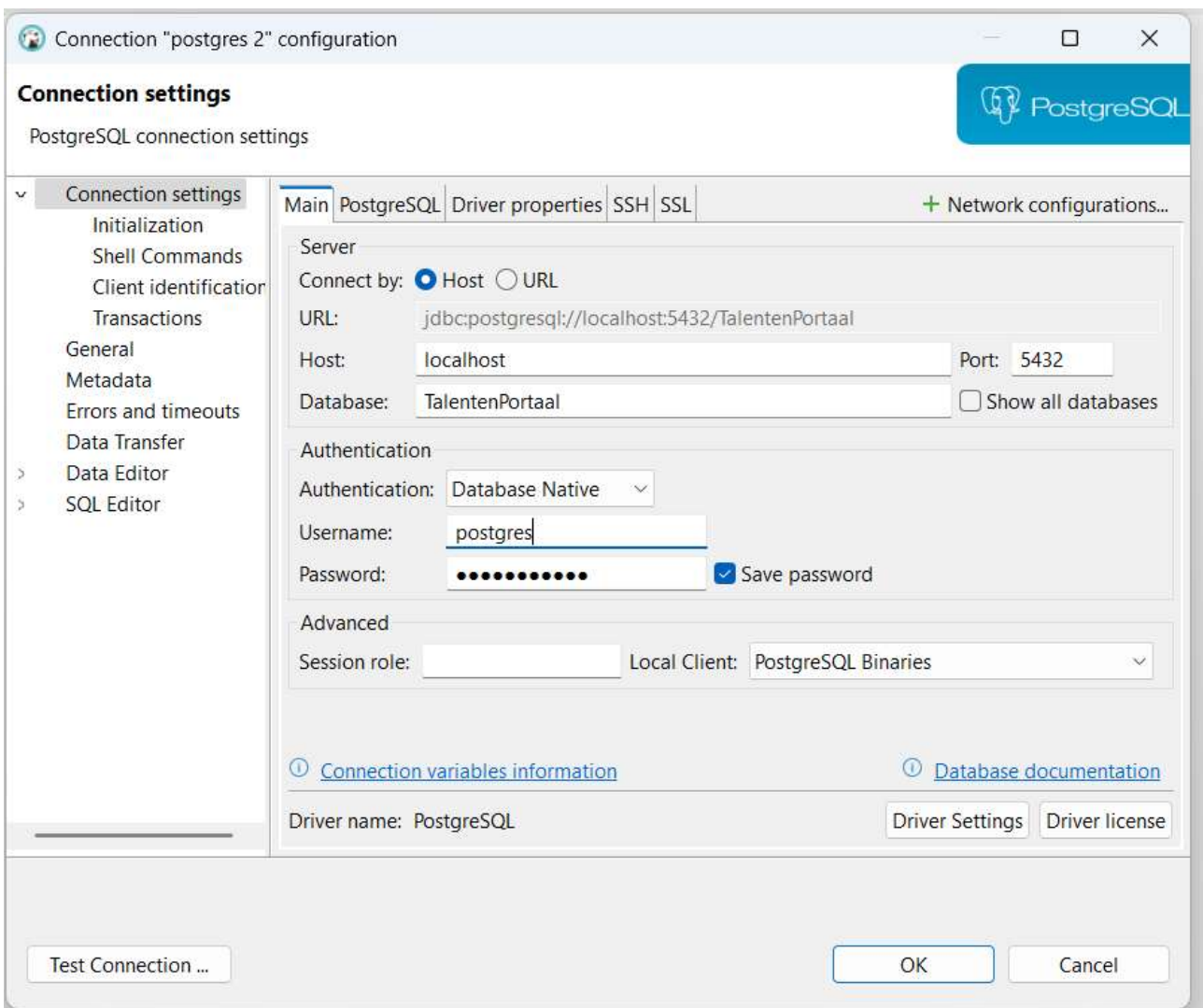
14. Start de DataSeeder



15. Wanneer de DataSeeder klaar is, dan wordt het bericht getoond:



16. Controleer de data in de PostgreSQL Database – bij voorkeur met het programma DBeaver – gebruik in ieder geval dezelfde credentials die eerder zijn gebruikt.



17. Iedere tabel bevat data. De database ziet er als volgt uit:

postgres 2	localhost:5432
Databases	
TalentenPortaal	
Schemas	
TalentenPortaal	
Tables	
Company	32K
EducationDetail	48K
Hobby	32K
Organization	32K
PastExperience	80K
Talent	48K
TalentPropertyLabel	48K
TalentPropertyMainCategoryLink	40K
TalentPropertySubCategoryLink	72K
VacancyComplete	112K
VacancyDemand	32K
VacancyDemandDetail	32K
VacancyOffer	32K
VacancyText	32K
Foreign Tables	
Views	
Materialized Views	
Indexes	
Functions	
Sequences	
Data types	
Aggregate functions	
public	

18. Online test-omgeving van Keycloak gebruiken.

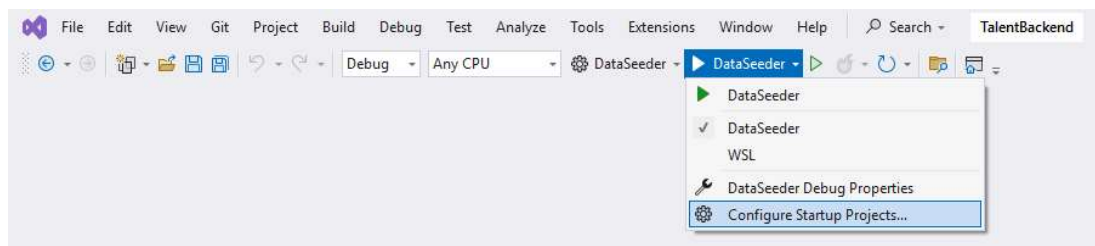
Zie hiervoor in Visual Studio 2022 opnieuw *appsettings.Development.json* in het bijzonder de sectie "Jwt". Zorg ervoor dat de juiste *key-value* paren beschikbaar zijn. Zie afbeelding hieronder:

```
"Jwt": {  
  "Authority": "https://bee-ids-test.azurewebsites.net/realms/bee-ideas-testing-realm",  
  "AuthorizationUrl": "https://bee-ids-test.azurewebsites.net/realms/bee-ideas-testing-realm/protocol/openid-connect/auth",  
  "TokenUrl": "https://bee-ids-test.azurewebsites.net/realms/bee-ideas-testing-realm/protocol/openid-connect/token",  
  "Audience": "account",  
  "Realm": "bee-ideas-testing-realm",  
  "ClientId": "api-core-client",  
  "ClientSecret": "tysyErCxjZYu0wGUMfv5jsbQt67LbM9a"  
},
```

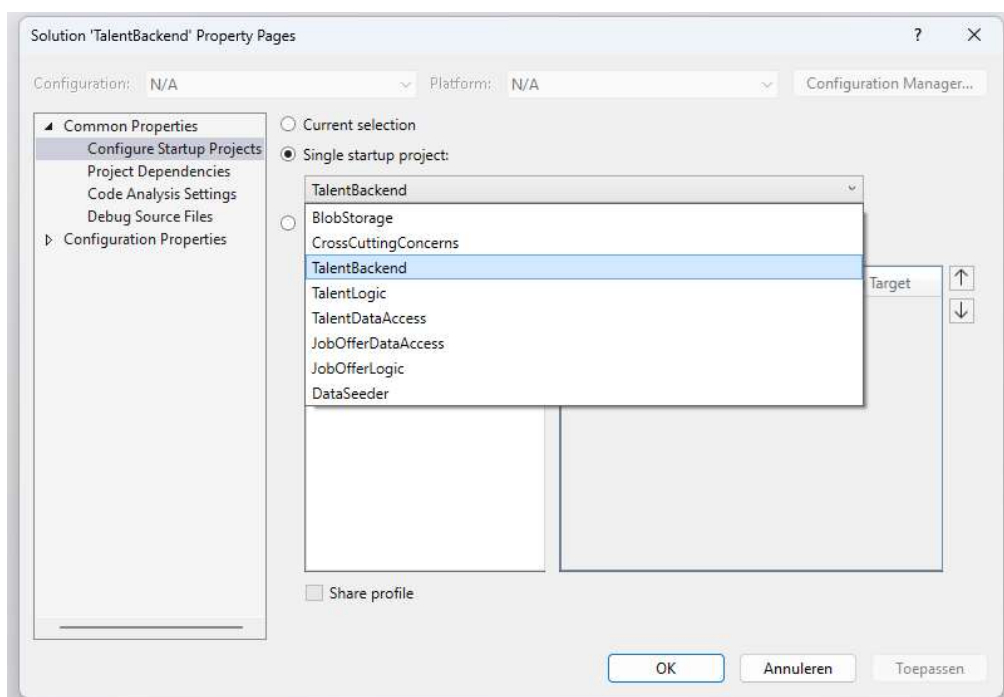
Afbeelding hierboven: de links die beginnen met *https* zijn beschikbaar en ook de juiste *clientSecret*. Alle links die beginnen met '<http://localhost:28080/>' kunnen verwijderd worden.

Opmerking: de bovenste *key-value* paren eindigen met een komma. Wanneer er geen komma is, dan leidt dat tot fouten. Alleen het laatste *key-value* paar hoeft geen komma op het einde te krijgen.

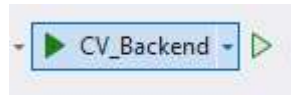
19. De WebAPI testen – Ga in Visual Studio 2022 opnieuw naar Configure Startup Projects



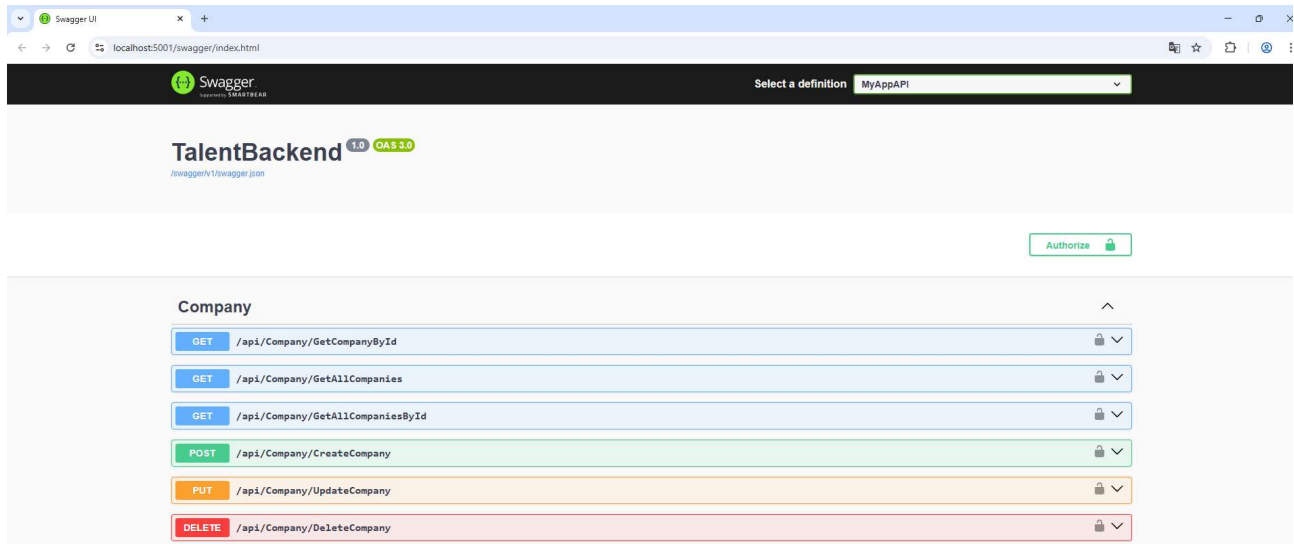
20. Selecteer TalentBackend en klik op OK.



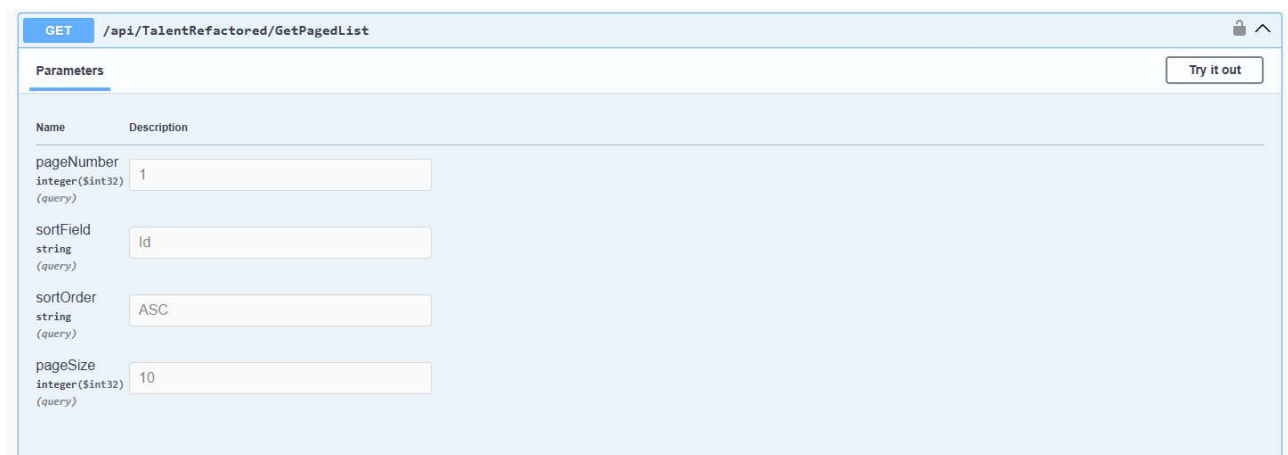
21. Start de CV_backend applicatie via de startknop bovenaan



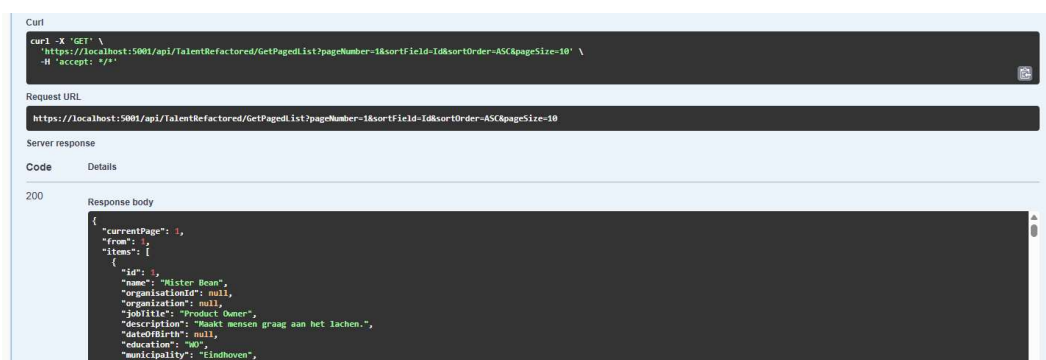
Na het opstarten wordt de *Swagger pagina* geopend.



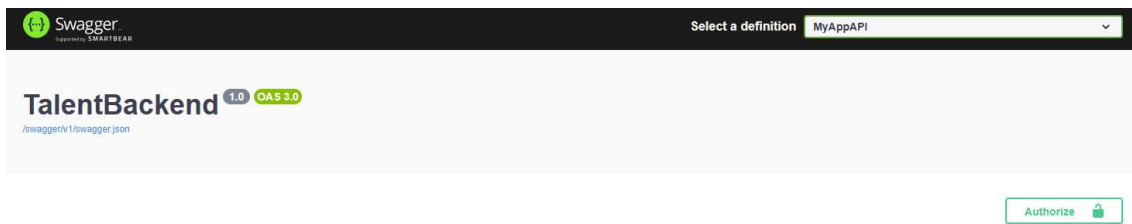
22. Scroll naar beneden naar *TalentRefactored*. Klap de route */api/TalentRefactored/GetPagedList* open. Klik op *Try it out*. Vul de gegevens in zoals in de afbeelding hieronder en klik op *Execute*.



Dit is een publiek toegankelijke route. Wanneer er code 200 wordt getoond, dan komt er succesvol data uit de database via de WebAPI.



23. Beschermd routes testen. Scroll op de *Swagger pagina* naar boven. Klik op de *Authorize button* rechts op het scherm.



24. Er wordt een pop-up geopend. Klik hier op *Authorize button*.

Available authorizations ✕

Scopes are used to grant an application different levels of access to data on behalf of the end user. Each API may declare one or more scopes.
API requires the following scopes. Select which ones you want to grant to Swagger UI.

Bearer (OAuth2, authorizationCode)

Application: KEYCLOAK

Authorization URL: <https://bee-ids-test.azurewebsites.net/realms/bee-ideas-testing-realm/protocol/openid-connect/auth>
Token URL: <https://bee-ids-test.azurewebsites.net/realms/bee-ideas-testing-realm/protocol/openid-connect/token>
Flow: authorizationCode

client_id:

client_secret:

25. Er volgt een redirect naar de Keycloak inlogpagina. Vul hier de volgende credentials in en klik op *Sign In*.


username: admin

password: testen



Sign in to your account

Username or email

Password
 

New user? [Register](#)

26. Er wordt teruggekomen op de *Swagger pagina*. Klik op de *Close* button.

Available authorizations

X

Scopes are used to grant an application different levels of access to data on behalf of the end user. Each API may declare one or more scopes. API requires the following scopes. Select which ones you want to grant to Swagger UI.

Bearer (OAuth2, authorizationCode)

Application: KEYCLOAK

Authorized

Authorization URL: <https://bee-ids-test.azurewebsites.net/realms/bee-ideas-testing-realm/protocol/openid-connect/auth>
Token URL: <https://bee-ids-test.azurewebsites.net/realms/bee-ideas-testing-realm/protocol/openid-connect/token>
Flow: authorizationCode
client_id: *****
client_secret: *****



Logout

Close

27. Scroll opnieuw naar beneden naar *TalentRefactored*. Klap de route `/api/TalentRefactored/GetAll` open. Klik op *Try it out* en klik vervolgens op *Execute*.

GET

/api/TalentRefactored/GetAll



Parameters

Cancel

No parameters

Execute

28. Wanneer deze beschermde route naar behoren werkt, dan wordt er code 200 teruggegeven voor OK. De teruggegeven data is in de *response body* te zien.

The screenshot displays a REST client interface with the following sections:

- Responses:** A dark-themed area at the top.
- Curl:** A text box containing the command:

```
curl -X 'GET' \  
  'https://localhost:5001/api/TalentRefactored/GetAll' \  
  -H 'accept: */*' \  
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpzZW50L3NpdCI6Im1hZC1lbnR5c2I6IjEwMDAwfQ0aM11joxNz02
```
- Request URL:** A text box containing the URL:

```
https://localhost:5001/api/TalentRefactored/GetAll
```
- Server response:** A section with a tab labeled 'Code' and a sub-tab 'Details'.
- Response body:** A text area showing a JSON response:

```
{  
  "description": "Technisch onderlegde en didactisch vaardige scrum master.Door diverse functies in software engineering ben ik technischonderlegd in allerlei takken van software ontwikkelen g. Tevensheb ik jarenlang als docent wiskunde voor de klas gestaamaarbij diverse sturende en motiverende vaardigheden belangrijkzijn. Tot mijn verrassing komen vele onderdelen uit deze functiesheel mooi samen in de rol van een scrum master. Ik wil nu graagalles combineren en inzetten om tot een mooi mogelijk resultaatte komen met een scrum team.",  
  "dateOfBirth": "1968-08-08",  
  "education": "HBO",  
  "municipality": "Grave",  
  "province": "Noord-Brabant",  
  "profilePhotoRef": "https://beedeasstorage.blob.core.windows.net/talentenportaal/images/jos.jpg",  
  "pitchUrl": "",  
  "tagId": "",  
  "cvUrl": "",  
  "linkedinUrl": "https://www.linkedin.com/in/jos-broeren4918833",  
  "contactOptions": "email, telefonisch, linkedIn",  
}
```

29. Naast *admin* zijn er ook de standaard users:

vacany_creator

user

registered

Het *password* van deze users is:

testen