# Prérequis

* .NET SDK
* Node.js et npm
* Angular CLI (Installé via npm : npm install -g @angular/cli)
* Docker

# Création d’un nouveau projet ASP.NET Core

dotnet new webapi -n MyBackend

cd MyBackend

dotnet run

# Création d’un nouveau projet Angular

ng new MyFrontend

cd MyFrontend

ng serve

# Integration entre Backend et Frontend

## Configurer les appels API

Modifier les services Angular pour appeler les API du backend ASP.NET Core. Utiliser HttpClient pour effectuer les requêtes HTTP.

## Configurer CORS

builder.Services.AddCors(options =>  
{  
 options.AddPolicy("Dev", policyBuilder =>  
 {  
 policyBuilder.WithOrigins("http://localhost:4200")  
 .AllowAnyMethod()  
 .AllowAnyHeader();  
 });  
});  
  
var app = builder.Build();  
app.UseCors("Dev");

## Installation de packages dans le backend via NuGet

* Microsoft.EntityFrameworkCore.SqlServer
* AutoMapper
* AutoMapper.Extensions.Microsoft.DependencyInjection
* System.Text.Json
* Swashbuckle.AspNetCore

## Configurer la chaîne de connection

"ConnectionStrings": {  
 "DefaultConnection": "Server=localhost;Database=MyDatabase;User Id=sa;Password=Your\_password123;"  
}

## Créer le contexte de la BD

public class ApplicationDbContext : DbContext  
{  
 public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)  
 : base(options)  
 {  
 }  
  
 public DbSet<YourEntity> YourEntities { get; set; }  
}

## Ajouter le contexte au service

builder.Services.AddDbContext< ApplicationDbContext>(cfg => cfg.UseSqlServer(  
 builder.Configuration.GetConnectionString("DefaultConnection")  
));

# Dockerfiles

## Backend

FROM mcr.microsoft.com/dotnet/aspnet:6.0 AS base  
WORKDIR /app  
EXPOSE 80  
  
FROM mcr.microsoft.com/dotnet/sdk:6.0 AS build  
WORKDIR /src  
COPY ["MyBackend/MyBackend.csproj", "MyBackend/"]  
RUN dotnet restore "MyBackend/MyBackend.csproj"  
COPY . .  
 WORKDIR "/src/MyBackend"  
RUN dotnet build "MyBackend.csproj" -c Release -o /app/build  
  
FROM build AS publish  
RUN dotnet publish "MyBackend.csproj" -c Release -o /app/publish  
  
FROM base AS final  
WORKDIR /app  
COPY --from=publish /app/publish .  
ENTRYPOINT ["dotnet", "MyBackend.dll"]

## Frontend

FROM node:14 AS build  
WORKDIR /app  
COPY . .  
RUN npm install  
RUN npm run build --prod  
  
FROM nginx:alpine  
COPY --from=build /app/dist/MyFrontend /usr/share/nginx/html  
EXPOSE 80  
ENTRYPOINT ["nginx", "-g", "daemon off;"]

# docker-compose.yml

version: '3.8'  
  
services:  
 sqlserver:  
 image: mcr.microsoft.com/mssql/server:2022-latest  
 container\_name: sqlserver  
 environment:  
 SA\_PASSWORD: "Your\_password123"  
 ACCEPT\_EULA: "Y"  
 ports:  
 - "1433:1433"  
  
backend:  
 build:  
 context: ./MyBackend  
 container\_name: backend  
 ports:  
 - "5000:80"  
 depends\_on:  
 - sqlserver  
 environment:  
 - ASPNETCORE\_ENVIRONMENT=Development  
  
frontend:  
 build:  
 context: ./MyFrontend  
 container\_name: frontend  
 ports:  
 - "4200:80"  
 depends\_on:  
 - backend