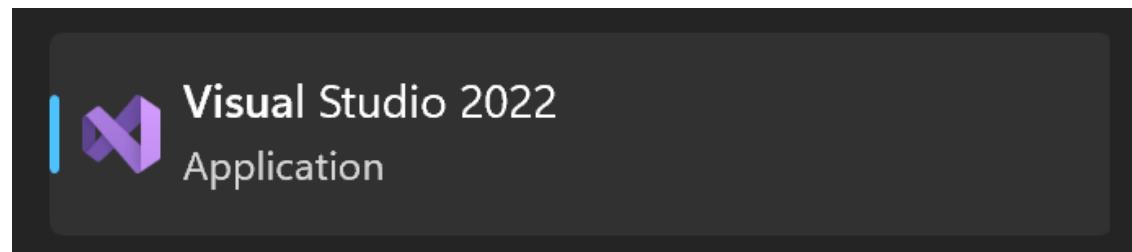


# Tutorial

How to create a fullstack web  
application using ASP.NET Core, React  
and Docker-Compose

1

Create solution



# Visual Studio 2022

## Ouvrir les éléments récents

Quand vous utilisez Visual Studio, tous les projets, dossiers ou fichiers que vous ouvrez s'affichent ici pour vous permettre d'y accéder rapidement.

Vous pouvez épinglez tout ce que vous ouvrez fréquemment pour l'afficher toujours en tête de liste.

## Démarrer



### Cloner un dépôt

Obtenir du code à partir d'un dépôt en ligne, par exemple GitHub ou Azure DevOps



### Ouvrir un projet ou une solution

Ouvrir un projet ou un fichier .sln Visual Studio local



### Ouvrir un dossier local

Naviguer parmi du code et le modifier dans n'importe quel dossier



### Créer un projet

Choisir un modèle de projet avec génération de modèles automatique de code pour bien démarrer

[Continuer sans code →](#)

2

Create *frontend* project

# Créer un projet

react

Tout effacer

## Modèles de projet récents

La liste des modèles récemment consultés sera affichée ici.

Tous les langages

Toutes les plateformes

Tous les types de projet



### Projet autonome JavaScript React

Un modèle de projet JavaScript **React** qui est démarré en lançant votre installation globale de npx.

JavaScript

Web



### Projet autonome TypeScript React

Un modèle de projet TypeScript **React** qui est démarré en lançant votre installation globale de npx.

TypeScript

Web



### ASP.NET Core avec React.js

Un modèle de projet pour créer une application ASP.NET Core avec **React.js**

C#

Linux

macOS

Windows

Cloud

Service

Web

Vous n'arrivez pas à trouver ce que vous cherchez ?

[Installer plus d'outils et de fonctionnalités](#)

Retour

Suivant

# Configurer votre nouveau projet

Projet autonome JavaScript React

JavaScript Web

Nom du projet

Emplacement



Nom de la solution

Placer la solution et le projet dans le même répertoire

Retour

Suivant

## Informations supplémentaires

Projet autonome JavaScript React

JavaScript Web

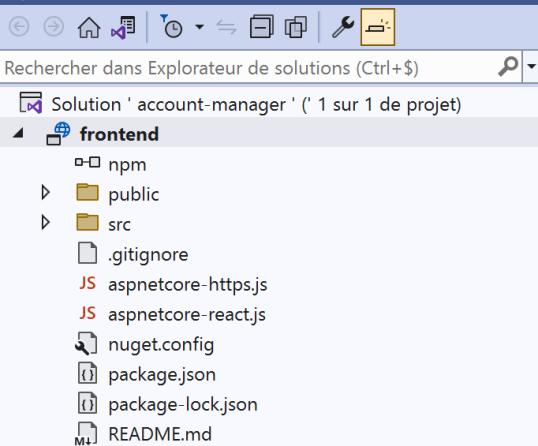
Add integration for Empty ASP.NET Web API project. 

[En savoir plus](#)

[Retour](#)

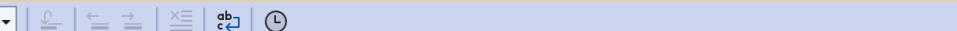
[Créer](#)

## Explorateur de solutions



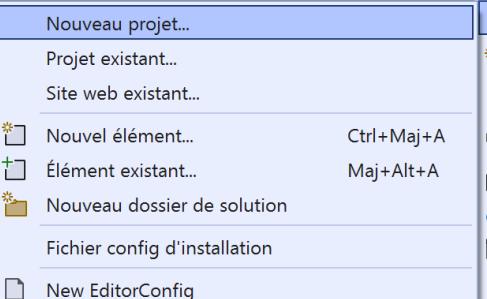
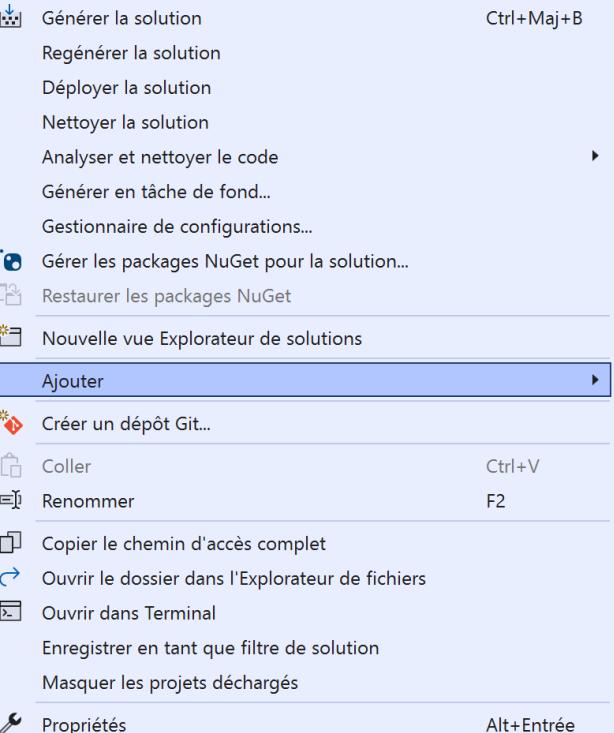
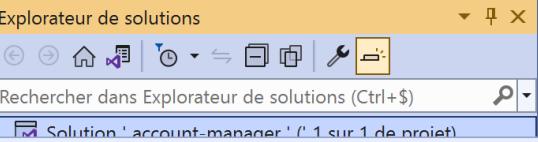
## Sortie

Afficher la sortie à partir de :



3

Create *backend* project



Sortie

Afficher la sortie à partir de :

PowerShell développeur Liste d'erreurs Sortie

Explorateur de solutions Modifications Git



# Ajouter un nouveau projet

api[Tout effacer](#)

## Modèles de projet récents

Projet autonome JavaScript React

JavaScript

Tous les langages



Toutes les plateformes



Tous les types de projet



### API web ASP.NET Core

Modèle de projet permettant de créer une application ASP.NET Core avec un exemple de contrôleur pour un service HTTP RESTful. Vous pouvez également utiliser ce modèle pour les vues et contrôleurs ASP.NET Core MVC.

C#

Linux

macOS

Windows

Cloud

Service

Web

WebAPI



### API web ASP.NET Core

Modèle de projet permettant de créer une application ASP.NET Core avec un exemple de contrôleur pour un service HTTP RESTful. Vous pouvez également utiliser ce modèle pour les vues et contrôleurs ASP.NET Core MVC.

F#

Linux

macOS

Windows

Cloud

Service

Web

WebAPI

Vous n'arrivez pas à trouver ce que vous cherchez ?

[Installer plus d'outils et de fonctionnalités](#)[Suivant](#)

# Configurer votre nouveau projet

API web ASP.NET Core

C#

Linux

macOS

Windows

Cloud

Service

Web

WebAPI

Nom du projet

backend

Emplacement

C:\Users\Kevin\source\repos\account-manager

...

Retour

Suivant

# Informations supplémentaires

## API web ASP.NET Core

C#

Linux

macOS

Windows

Cloud

Service

Web

WebAPI

### Infrastructure i

.NET 7.0 (Prise en charge des termes standard)

### Type d'authentification i

Aucun

Configurer pour HTTPS i

Activer Docker i

### Docker OS i

Linux

Utiliser des contrôleurs (décocher pour utiliser un minimum d'API) i

Activer la prise en charge d'OpenAPI i

N'utilisez pas d'instructions de niveau supérieur. i

Retour

Créer

# 4

*Setup docker-compose project*

Fichier Edition Affichage Git Projet Générer Déboguer Test Analyser Outils Extensions Fenêtre Aide Rechercher (Ctrl+Q) account manager Connexion Live Share

backend: Vue d'ensemble | Debug Any CPU localhost (Edge)

Vue d'ensemble Services connectés Publier

# ASP.NET Core

Découvrez la plateforme .NET, créez votre première application.

Générer votre application

Parcourir les documentations, les exemples et les didacticiels

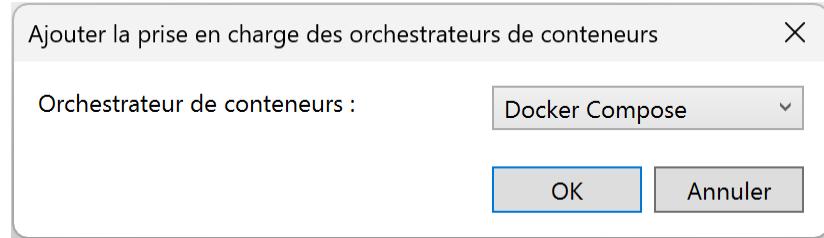
Architecture d'application .NET

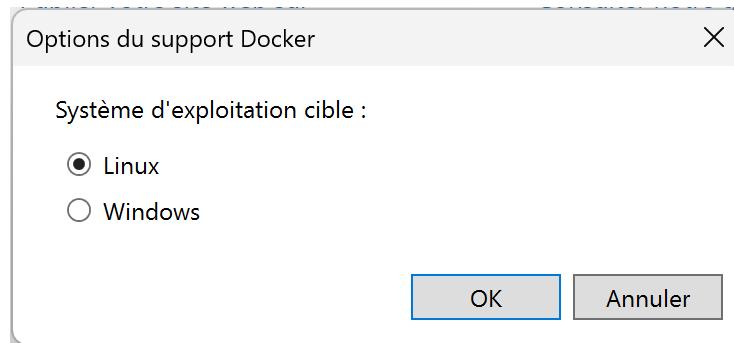
Sortie

Afficher la sortie à partir de : Dépendances de service

Nouvel élément... Ctrl+Maj+A  
Élément existant... Maj+Alt+A  
Nouvel élément généré automatiquement...  
Nouveau dossier  
Prise en charge des orchestrateurs de conteneurs  
Prise en charge de Docker...  
Application Insights Telemetry...  
Bibliothèque côté client...  
Modèle de Machine Learning...  
Nouveau projet de tâche web Azure  
Projet existant en tant que tâche web Azure  
Référence de projet...  
Référence de projet partagé...  
Référence COM...  
Référence de service...  
Service connecté  
Classe...  
New EditorConfig  
Nouvel EditorConfig (IntelliCode)

Générer Regénérer Nettoyer Affichage Analyser et nettoyer le code Compresser Publier... Configurer Application Insights... Vue d'ensemble Limiter à ceci Nouvelle vue Explorateur de solutions Imbrication de fichiers Modifier le fichier projet Dépendances de build Ajouter Gérer les packages NuGet... Gérer les bibliothèques côté client... Gérer les données secrètes de l'utilisateur Supprimer les références inutilisées... Synchroniser les espaces de noms Définir en tant que projet de démarrage Déboguer Couper Ctrl+X Supprimer F2 Renommer F2 Décharger le projet Charger les dépendances directes Charger l'ensemble de l'arborescence des dépendances Copier le chemin d'accès complet Ouvrir le dossier dans l'Explorateur de fichiers Ouvrir dans Terminal Propriétés Alt+Entrée





The screenshot shows a Visual Studio interface with the following components:

- Top Bar:** Fichier, Edition, Affichage, Git, Projet, Générer, Déboguer, Test, Analyser, Outils, Extensions, Fenêtre, Aide, Rechercher (Ctrl+Q), account-manager, Connexion.
- Toolbar:** Standard icons for file operations like Open, Save, Print, etc.
- Project Selector:** Debug, Any CPU, docker-compose.
- Status Bar:** PowerShell développeur, Liste d'erreurs, Sortie, Explorateur de solutions, Modifications Git.
- Code Editor:** backend: Vue d'ensemble, displaying a Docker Compose configuration file (docker-compose.override.yml). The code includes XML tags for Project Version, Docker Target OS, Project GUID, Docker Launch Action, Docker Service URL, and Docker Service Name.
- Solution Explorer:** Explorateur de solutions, showing the solution structure with three projects: backend, docker-compose, and frontend.
- Output Window:** Sortie, showing "Aucun problème détecté".
- Bottom Status Bar:** Prêt, Ajouter au contrôle de code source, Sélectionner le référentiel.

frontend

Nouveau Trier Afficher

Kevin Doolaege > source > repos > account-manager > frontend

Rechercher dans : frontend

Nom	Modifié le	Type	Taille
.vscode	11/02/2023 17:50	Dossier de fichiers	
node_modules	11/02/2023 17:50	Dossier de fichiers	
public	11/02/2023 17:49	Dossier de fichiers	
src	11/02/2023 17:50	Dossier de fichiers	
.gitignore	11/02/2023 17:49	Fichier GITIGNORE	1 Ko
aspnetcore-https.js	11/02/2023 17:50	JSFile	2 Ko
aspnetcore-react.js	11/02/2023 17:50	JSFile	2 Ko
<input checked="" type="checkbox"/> Dockerfile	11/02/2023 19:54	Fichier	0 Ko
frontend.esproj	11/02/2023 17:50	Fichier ESPROJ	1 Ko
nuget.config	11/02/2023 17:50	XML Configuration Fi...	1 Ko
package.json	11/02/2023 17:50	Fichier JSON	2 Ko
package-lock.json	11/02/2023 17:50	Fichier JSON	1209 Ko
README.md	11/02/2023 17:49	Fichier MD	4 Ko

Accueil

OneDrive - Personal

Ce PC

NAS605B22

Local Disk (C:)

Bureau

Téléchargements

Documents

Images

Kevin

Ce PC

Dossier personnel (NAS605B2)

Multimédia (NAS605B22)

Local Disk (C:)

Bibliothèques

Réseau

Linux

13 élément(s) | 1 élément sélectionné 0 octet(s) |

\*Dockerfile - Bloc-notes

Fichier   Modifier   Affichage  

```
# Stage 1: Compile and build
FROM node:alpine AS build
WORKDIR /app
COPY frontend/package*.json .
RUN npm install
COPY frontend/
RUN npm run build

# Stage 2: Serve app with nginx server
FROM nginx:alpine
WORKDIR /usr/share/nginx/html
COPY --from=build /app/build .
EXPOSE 80
ENTRYPOINT [ "nginx", "-g", "daemon off;" ]
```

Ln 14, Col 44 | 100% | Windows (CRLF) | UTF-8

The screenshot shows the Visual Studio interface with the following components:

- Top Bar:** Fichier, Edition, Affichage, Git, Projet, Générer, Déboguer, Test, Analyser, Outils, Extensions, Fenêtre, Aide, Rechercher (Ctrl+Q), account-manager, Connexion.
- Toolbar:** Standard icons for file operations like Open, Save, Print, etc.
- Project Explorer:** Shows the solution structure:
  - Solution 'account-manager' (3 sur 3 de projets)
    - backend
      - Connected Services
      - Dépendances
      - Properties
      - Controllers
      - appsettings.json
      - Dockerfile
      - C# Program.cs
      - C# WeatherForecast.cs
    - docker-compose
      - .dockerignore
      - docker-compose.yml
    - frontend
      - npm
      - public
      - src
        - .gitignore
        - JS aspnetcore-https.js
        - JS aspnetcore-react.js
        - Dockerfile
        - nuget.config
        - package.json
        - package-lock.json
- Code Editor:** docker-compose.yml (services section) with syntax highlighting for YAML and Dockerfile sections.
- Status Bar:** 100 %, Aucun problème détecté, Lig. : 24, Car. : 12, SPC, CRLF.
- Output Window:** Sortie, Afficher la sortie à partir de : Outils de conteneur, showing errors related to Docker Desktop.
- Bottom Navigation:** Explorateur de solutions, Modifications Git.

```
version: '3.4'

services:
  backend:
    container_name: backend
    build:
      context: .
      dockerfile: backend/Dockerfile
    restart: always
    depends_on:
      - database
    networks:
      - local

  frontend:
    container_name: frontend
    build:
      context: .
      dockerfile: frontend/Dockerfile
    restart: always
    networks:
      - local

  database:
    container_name: database
    image: mariadb
    restart: always
    volumes:
      - data:/var/lib/mysql
      - ./database/initdb.sql:/docker-entrypoint-initdb.d/initdb.sql
    environment:
      - MYSQL_ROOT_PASSWORD=gloplop
      - MYSQL_DATABASE=db
      - MYSQL_USER=pifou
      - MYSQL_PASSWORD=pasglop
    networks:
      - local

  reverse-proxy:
    container_name: reverse-proxy
    build:
      context: .
      dockerfile: reverse-proxy/Dockerfile
    restart: always
    ports:
      - "80:80"
      - "443:443"
    networks:
      - local

  networks:
    local:

  volumes:
    data:
```

# 6

Add *database* instance

account-manager

Nouveau | Trier | Afficher | ...

Kevin Doolaege > source > repos > account-manager

Rechercher dans : account-manager

Accueil

OneDrive - Personal

Ce PC

NAS605B22

Local Disk (C:)

Bureau

Téléchargements

Documents

Images

Kevin

Ce PC

Dossier personnel (NAS605B2)

Multimédia (NAS605B22)

Local Disk (C:)

Bibliothèques

Réseau

Linux

Nom

Modifié le

Type

Taille

backend

11/02/2023 17:54

Dossier de fichiers

frontend

11/02/2023 17:50

Dossier de fichiers

obj

11/02/2023 17:54

Dossier de fichiers

.dockernignore

11/02/2023 17:54

Fichier DOCKERIGNO...

1 Ko

account-manager.sln

11/02/2023 17:54

Visual Studio Solution

3 Ko

docker-compose.dcproj

11/02/2023 17:54

docker-compose.override.yml

11/02/2023 17:54

docker-compose.yml

11/02/2023 17:54

Dossier

Raccourci

Microsoft Access Database

Document Microsoft Word

Microsoft Access Database

Présentation Microsoft PowerPoint

Microsoft Publisher Document

Document texte

Feuille de calcul Microsoft Excel

Dossier compressé

Affichage

Trier par

Regrouper par

Annuler la suppression Ctrl+Z

Nouveau

Propriétés Alt+Entrée

Ouvrir dans le Terminal

Afficher plus d'options Maj+F10

8 élément(s)

account-manager

Nouveau | Trier | Afficher | ...

Kevin Doolaege > source > repos > account-manager

Rechercher dans : account-manager

Accueil

OneDrive - Personal

Ce PC

NAS605B22

Local Disk (C:)

Bureau

Téléchargements

Documents

Images

Kevin

Ce PC

Dossier personnel (NAS605B2)

Multimédia (NAS605B22)

Local Disk (C:)

Bibliothèques

Réseau

Linux

Nom | Modifié le | Type | Taille

Nom	Modifié le	Type	Taille
backend	11/02/2023 17:54	Dossier de fichiers	
database	11/02/2023 17:55	Dossier de fichiers	
frontend	11/02/2023 17:50	Dossier de fichiers	
obj	11/02/2023 17:54	Dossier de fichiers	
.dockerignore	11/02/2023 17:54	Fichier DOCKERIGNO...	1 Ko
account-manager.sln	11/02/2023 17:54	Visual Studio Solution	3 Ko
docker-compose.dcproj	11/02/2023 17:54	Fichier DCPROJ	1 Ko
docker-compose.override.yml	11/02/2023 17:54	Fichier YML	1 Ko
docker-compose.yml	11/02/2023 17:58	Fichier YML	1 Ko

9 élément(s) |

database

Nouveau ▾ ⌂ ⌂ ⌂ ⌂ Trier ▾ Afficher ▾ ...

← → ↑ ⌂ Kevin Doolaege > source > repos > account-manager > database ⌂ Rechercher dans : database

Accueil

OneDrive - Personal

Ce PC

NAS605B22

Local Disk (C:)

Bureau

Téléchargements

Documents

Images

Kevin

Ce PC

Dossier personnel (NAS605B2)

Multimédia (NAS605B22)

Local Disk (C:)

Bibliothèques

Réseau

Linux

1 élément | 1 élément sélectionné 0 octet(s) |

Nom	Modifié le	Type	Taille
initdb.sql	11/02/2023 19:45	Fichier SQL	0 Ko



Fichier Modifier Affichage

```
CREATE DATABASE IF NOT EXISTS db;
USE db;

CREATE TABLE IF NOT EXISTS Accounts
(
    Id int PRIMARY KEY NOT NULL AUTO_INCREMENT,
    Name varchar(100)
);

CREATE TABLE IF NOT EXISTS Users
(
    Id int PRIMARY KEY NOT NULL AUTO_INCREMENT,
    Name varchar(100),
    Email varchar(100),
    Password varchar(100)
);
```

# 7

*Link backend to database instance*

Fichier Edition Affichage Git Projet Générer Déboguer XML Test Analyser Outils Extensions Fenêtre Aide Rechercher (Ctrl+Q) account-manager Connexion Live Share

backend.csproj

```
<Project Sdk="Microsoft.NET.Sdk.Web">
  <PropertyGroup>
    <TargetFramework>net7.0</TargetFramework>
    <Nullable>enable</Nullable>
    <ImplicitUsings>enable</ImplicitUsings>
    <DockerDefaultTargetOS>Linux</DockerDefaultTargetOS>
    <DockerComposeProjectPath>..\docker-compose.dcproj</DockerComposeProjectPath>
  </PropertyGroup>
  <ItemGroup>
    <PackageReference Include="Microsoft.AspNetCore.OpenApi" Version="7.0.1" />
    <PackageReference Include="Microsoft.VisualStudio.Azure.Containers.Tools.Targets" Version="1.17.0" />
    <PackageReference Include="Swashbuckle.AspNetCore" Version="6.4.0" />
  </ItemGroup>
</Project>
```

Aucun problème détecté

Sortie

Afficher la sortie à partir de : Outils de conteneur

Impossible de déterminer les images à tirer (pull).  
Démarrage de Docker Desktop...  
Docker Desktop ne fonctionne pas comme prévu. Essayez de redémarrer Docker.

Explorateur de solutions

Solution 'account-manager' (3 sur 3 de projets)

- backend
  - Connected Services
  - Dépendances
  - Properties
  - Controllers
  - appsettings.json
  - Dockerfile
  - C# Program.cs
  - C# WeatherForecast.cs
- docker-compose
  - .dockerignore
  - docker-compose.yml
- frontend
  - npm
  - public
  - src
    - .gitignore
    - aspnetcore-https.js
    - aspnetcore-react.js
    - Dockerfile
    - nuget.config
    - package.json
    - package-lock.json
    - README.md

Explorateur de solutions Modifications Git

The screenshot shows the Visual Studio IDE interface with the following details:

- Top Bar:** Fichier, Edition, Affichage, Git, Projet, Générer, Déboguer, XML, Test, Analyser, Outils, Extensions, Fenêtre, Aide, Rechercher (Ctrl+Q), account-manager, Connexion.
- Toolbar:** Standard icons for file operations like Open, Save, Print, etc.
- Project Selector:** Debug, Any CPU, docker-compose.
- Status Bar:** Docker Compose, Live Share.
- Code Editor:** The 'backend' project file is open, showing .NET Core configuration. Lines 12-15 are selected:

```
<ItemGroup>
    <PackageReference Include="Microsoft.AspNetCore.OpenApi" Version="7.0.2" />
    <PackageReference Include="Microsoft.VisualStudio.Azure.Containers.Tools.Targets" Version="1.17.0" />
    <PackageReference Include="Pomelo.EntityFrameworkCore.MySql" Version="7.0.0" />
    <PackageReference Include="Swashbuckle.AspNetCore" Version="6.5.0" />
```
- Solution Explorer:** Shows three projects: backend, docker-compose, and frontend. The backend project contains files like Connected Services, Dépendances, Properties, Controllers, appsettings.json, Dockerfile, Program.cs, WeatherForecast.cs. The docker-compose project contains .dockerignore and docker-compose.yml. The frontend project contains npm, public, src, .gitignore, aspnetcore-https.js, aspnetcore-react.js, Dockerfile, nuget.config, package.json, package-lock.json, and README.md.
- Output Window:** Displays the message "Aucun problème détecté" (No errors detected).
- Status Bar:** Shows the current file is 'backend', and other status indicators like 'Ajouter au contrôle de code source' (Add to code control) and 'Sélectionner le référentiel' (Select the repository).

```
backend* ▾ X
1 <Project Sdk="Microsoft.NET.Sdk.Web">
2
3   <PropertyGroup>
4     <TargetFramework>net7.0</TargetFramework>
5     <Nullable>enable</Nullable>
6     <ImplicitUsings>enable</ImplicitUsings>
7     <DockerDefaultTargetOS>Linux</DockerDefaultTargetOS>
8     <DockerComposeProjectPath>..\docker-compose.dcproj</DockerComposeProjectPath>
9   </PropertyGroup>
10
11   <ItemGroup>
12     <PackageReference Include="Microsoft.AspNetCore.OpenApi" Version="7.0.2" />
13     <PackageReference Include="Microsoft.VisualStudio.Azure.Containers.Tools.Targets" Version="1.17.0" />
14     <PackageReference Include="Pomelo.EntityFrameworkCore.MySql" Version="7.0.0" />
15     <PackageReference Include="Swashbuckle.AspNetCore" Version="6.5.0" />
16   </ItemGroup>
17
18 </Project>
19
```

100 % Aucun problème détecté

## Sortie

Afficher la sortie à partir de : Outils de conteneur

Impossible de déterminer les images à tirer (pull).

Démarrage de Docker Desktop...

Docker Desktop ne fonctionne pas comme prévu. Essayez de redémarrer Docker.

Explorateur de solutions

Rechercher dans Explorateur de solutions (Ctrl+\$)

- ▲ Générer
- Regénérer
- Nettoyer
- Affichage
- Analyser et nettoyer le code
- Compresser
- Publier...
- Configurer Application Insights...
- Vue d'ensemble
- Limiter à ceci
- Nouvelle vue Explorateur de solutions
- Imbrication de fichiers
- ↶ Modifier le fichier projet
- Dépendances de build
- Ajouter
- 🌐 Gérer les packages NuGet...
- 🌐 Gérer les bibliothèques côté client...
- Gérer les données secrètes de l'utilisateur
- Supprimer les références inutilisées...
- Synchroniser les espaces de noms
- ⚙ Définir en tant que projet de démarrage
- Déboguer
- ✖ Couper Ctrl+X
- ✗ Supprimer Suppr
- renommer F2
- Décharger le projet
- Charger les dépendances directes
- Charger l'ensemble de l'arborescence des dépendances
- Copier le chemin d'accès complet
- ↶ Ouvrir le dossier dans l'Explorateur de fichiers
- Ouvrir dans Terminal
- 🔧 Propriétés Alt+Entrée

Fichier Edition Affichage Git Projet Générer Déboguer XML Test Analyser Outils Extensions Fenêtre Aide Rechercher (Ctrl+Q) account-manager Connexion Live Share

backend\* | Debug Any CPU docker-compose

Nouvel élément... Ctrl+Maj+A  
Élément existant... Maj+Alt+A  
Nouvel élément généré automatiquement...  
Nouveau dossier  
Prise en charge de Docker...  
Application Insights Telemetry...  
Bibliothèque côté client...  
Modèle de Machine Learning...  
Nouveau projet de tâche web Azure  
Projet existant en tant que tâche web Azure  
Référence de projet...  
Référence de projet partagé...  
Référence COM...  
Référence de service...  
Service connecté  
Classe...  
New EditorConfig  
Nouvel EditorConfig (IntelliCode)

Explorateur de solutions

Générer  
Regénérer  
Nettoyer  
Affichage  
Analyser et nettoyer le code  
Compresser  
Publier...  
Configurer Application Insights...  
Vue d'ensemble  
Limiter à ceci  
Nouvelle vue Explorateur de solutions  
Imbrication de fichiers  
Modifier le fichier projet  
Dépendances de build  
Ajouter  
Gérer les packages NuGet...  
Gérer les bibliothèques côté client...  
Gérer les données secrètes de l'utilisateur  
Supprimer les références inutilisées...  
Synchroniser les espaces de noms  
Définir en tant que projet de démarrage  
Déboguer  
Couper Ctrl+X  
Supprimer Suppr  
Renommer F2  
Décharger le projet  
Charger les dépendances directes  
Charger l'ensemble de l'arborescence des dépendances  
Copier le chemin d'accès complet  
Ouvrir le dossier dans l'Explorateur de fichiers  
Ouvrir dans Terminal  
Propriétés Alt+Entrée

100 % Aucun problème détecté Lig. : 18 Car. : 11

Sortie

Afficher la sortie à partir de : Outils de conteneur

Impossible de déterminer les images à tirer (pull).  
Démarrage de Docker Desktop...  
Docker Desktop ne fonctionne pas comme prévu. Essayez de redémarrer Docker.

Prêt ↑ A

The screenshot shows the Visual Studio IDE interface with the following details:

- Top Bar:** Fichier, Edition, Affichage, Git, Projet, Générer, Déboguer, Test, Analyser, Outils, Extensions, Fenêtre, Aide, Rechercher (Ctrl+Q), account-manager, Connexion.
- Toolbar:** Standard icons for file operations like Open, Save, Print, etc.
- Project Explorer:** Shows the solution structure with three projects: backend, docker-compose, and frontend.
- Code Editor:** The current file is "backend.csproj". The code defines a Microsoft.NET.Sdk.Web project targeting net7.0. It includes a PropertyGroup for DockerDefaultTargetOS set to Linux and DockerComposeProjectPath set to ..\docker-compose.dcproj. It also lists several package references including Microsoft.AspNetCore.OpenApi, Microsoft.VisualStudio.Azure.Containers.Tools.Targets, Pomelo.EntityFrameworkCore.MySql, Swashbuckle.AspNetCore, Microsoft.VisualStudio.Azure.Containers.Tools.Targets, and Pomelo.EntityFrameworkCore.MySql.
- Solution Explorer:** Shows the contents of the backend project, including Connected Services, Dépendances, Properties, Controllers, and Services (selected). It also lists Dockerfile, appsettings.json, Program.cs, WeatherForecast.cs, and other files.
- Output Window:** Displays the following text:
  - Aucun problème détecté
  - Impossible de déterminer les images à tirer (pull).
  - Démarrage de Docker Desktop...
  - Docker Desktop ne fonctionne pas comme prévu. Essayez de redémarrer Docker.
- Bottom Status Bar:** Cet élément ne prend pas en charge la fonction d'aperçu, Ajouter au contrôle de code source, Sélectionner le référentiel, and a Git icon.

Eichier Edition Affichage Git Projet Générer Déboguer Test Analyser Outils Extensions Fenêtre Aide Rechercher (Ctrl+Q) account-manager Connexion Live Share

backend\* Debug Any CPU docker-compose Docker Compose

`<Project Sdk="Microsoft.NET.Sdk.Web">`

`<PropertyGroup>`

`<TargetFramework>net7.0</TargetFramework>`

`<Nullable>enable</Nullable>`

`<ImplicitUsings>enable</ImplicitUsings>`

`<DockerDefaultTargetOS>Linux</DockerDefaultTargetOS>`

`<DockerComposeProjectPath>..\docker-compose.dcproj</DockerComposeProjectPath>`

`</PropertyGroup>`

`<ItemGroup>`

`<PackageReference Include="Microsoft.AspNetCore.OpenApi" Version="7.0.2" />`

`<PackageReference Include="Microsoft.VisualStudio.Azure.Containers.Tools.Targets" Version="1.17.0" />`

`<PackageReference Include="Pomelo.EntityFrameworkCore.MySql" Version="7.0.0" />`

`<PackageReference Include="Swashbuckle.AspNetCore" Version="6.5.0" />`

`<PackageReference Include="Microsoft.VisualStudio.Azure.Container" />`

`<PackageReference Include="Pomelo.EntityFrameworkCore.MySql" Versi`

`</ItemGroup>`

`<ItemGroup>`

`<Folder Include="Services\" />`

`</ItemGroup>`

`</Project>`

100 % Aucun problème détecté

Sortie

Afficher la sortie à partir de : Outils de conteneur

Impossible de déterminer les images à tirer (pull).  
Démarrage de Docker Desktop...  
Docker Desktop ne fonctionne pas comme prévu. Essayez de redémarrer Docker.

Explorateur de solutions

Rechercher dans Explorateur de solutions (Ctrl+\$)

Solution 'account-manager' (3 sur 3 de projets)

backend

- Connected Services
- Dépendances
- Properties
- Controllers
- Services

Afficher dans le navigateur (Microsoft Edge) Ctrl+Maj+W  
Naviguer avec...

Ajouter

- Nouvel élément... Ctrl+Maj+A
- Élément existant... Maj+Alt+A
- Nouvel élément généré automatiquement...
- Nouveau dossier
- Application Insights Telemetry...
- Bibliothèque côté client...
- Modèle de Machine Learning...
- Nouveau projet de tâche web Azure
- Projet existant en tant que tâche web Azure
- Classe...
- New EditorConfig
- Nouvel EditorConfig (IntelliCode)

REN README.md

Explorateur de solutions Modifications Git

Prêt Ajouter au contrôle de code source Sélectionner le référentiel

## Ajouter un nouvel élément - backend

? X

## Installé

Trier par : Par défaut



Rechercher (Ctrl+E)

## C#

- Général
- ASP.NET Core

## En ligne

	Classe	C#
	Interface	C#
	Composant Razor	C#
	Contrôleur MVC avec des actions de lecture/écriture	C#
	Contrôleur MVC - Vide	C#
	Contrôleur d'API - Vide	C#
	Contrôleur d'API avec actions de lecture/écriture	C#
	Page Razor - Vide	C#
	Vue Razor - Vide	C#
	Disposition Razor	C#
	Début de la vue Razor	C#
	Fichier de code	C#
	Fichier d'informations de l'assembly	C#
	Machine Learning Model (ML.NET)	C#

Type : C#

Déclaration de classe vide

Nom :

DbContext.cs

Ajouter

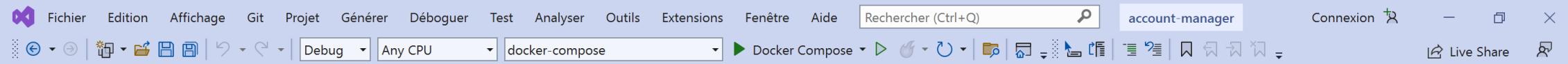
Annuler

The screenshot shows the Visual Studio IDE interface with the following components:

- Top Bar:** Fichier, Edition, Affichage, Git, Projet, Générer, Déboguer, Test, Analyser, Outils, Extensions, Fenêtre, Aide, Rechercher (Ctrl+Q), account-manager, Connexion.
- Toolbar:** Includes icons for Undo, Redo, Save, Open, Print, and various project-related functions.
- Project Explorer:** Shows the solution structure with three projects: backend, docker-compose, and frontend.
- Code Editor:** Displays the `backend/Services/DatabaseContext.cs` file content:

```
1  namespace backend.Services {  
2      public class DatabaseContext {  
3      }  
4  }  
5
```
- Output Window:** Shows the following text:

```
Aucun problème détecté  
Afficher la sortie à partir de : Outils de conteneur  
Impossible de déterminer les images à tirer (pull).  
Démarrage de Docker Desktop...  
Docker Desktop ne fonctionne pas comme prévu. Essayez de redémarrer Docker.
```
- Solution Explorer:** Shows the contents of each project folder.
- Bottom Bar:** Includes buttons for "Prêt", "Ajouter au contrôle de code source", "Selectionner le référentiel", and a status bar indicating "2" changes.



DatabaseContext.cs    backend\*

backend

```
1  using backend.Models;
2  using Microsoft.EntityFrameworkCore;
3  using System.Collections.Generic;
4  using System.Security.Principal;
5
6  namespace backend.Services {
7
8      public class DatabaseContext : DbContext {
9
10         public DatabaseContext(DbContextOptions<DatabaseContext> options) : base(options) { }
11
12         public DbSet<User> Users { get; set; } = null!;
13     }
14 }
```

Explorateur de solutions

Rechercher dans Explorateur de solutions (Ctrl+\$)

- Solution 'account-manager' (3 sur 3 de projets)
  - backend
    - Connected Services
    - Dépendances
    - Properties
    - Controllers
    - Services
      - DbContext.cs
      - appsettings.json
      - Dockerfile
      - Program.cs
      - WeatherForecast.cs
  - docker-compose
    - .dockerignore
    - docker-compose.yml
  - frontend
    - npm
    - public
    - src
      - .gitignore
      - aspnetcore-https.js
      - aspnetcore-react.js
      - Dockerfile
      - nuget.config
      - package.json
      - package-lock.json
      - README.md

Sortie

Aucun problème détecté

Impossible de déterminer les images à tirer (pull).  
Démarrage de Docker Desktop...  
Docker Desktop ne fonctionne pas comme prévu. Essayez de redémarrer Docker.

Explorateur de solutions    Modifications Git

The screenshot shows the Visual Studio interface with the following details:

- Top Bar:** Fichier, Edition, Affichage, Git, Projet, Générer, Déboguer, Test, Analyser, Outils, Extensions, Fenêtre, Aide, Rechercher (Ctrl+Q), account-manager, Connexion.
- Toolbar:** Includes icons for Undo, Redo, Save, Open, Print, and various project management tools.
- Current Project:** docker-compose (Any CPU).
- Solution Explorer:** Shows the solution structure with three projects:
  - backend:** Contains Connected Services, Dépendances, Properties, Controllers, Services, appsettings.json, Dockerfile, Program.cs, WeatherForecast.cs.
  - docker-compose:** Contains .dockerignore, docker-compose.yml.
  - frontend:** Contains npm, public, src, .gitignore, aspnetcore-https.js, aspnetcore-react.js, Dockerfile, nuget.config, package.json, package-lock.json, README.md.
- Code Editor:** Displays the content of the 'appsettings.json' file, which defines logging levels and allowed hosts.
- Status Bar:** 100%, Aucun problème détecté, Lig. : 10, Car. : 1, SPC, CRLF, Explorateur de solutions, Modifications Git.
- Bottom Status Bar:** Prêt, Ajouter au contrôle de code source, Sélectionner le référentiel, 2.

```
1 {  
2     "Logging": {  
3         "LogLevel": {  
4             "Default": "Information",  
5             "Microsoft.AspNetCore": "Warning"  
6         }  
7     },  
8     "AllowedHosts": "*"  
9 }  
10
```

The screenshot shows the Visual Studio IDE interface with the following details:

- Top Bar:** Fichier, Edition, Affichage, Git, Projet, Générer, Déboguer, Test, Analyser, Outils, Extensions, Fenêtre, Aide, Rechercher (Ctrl+Q), account-manager, Connexion.
- Toolbar:** Includes icons for Undo, Redo, Save, Open, Print, and various project management tools.
- Current Project:** docker-compose (Any CPU).
- Solution Explorer:** Shows the solution structure with three projects: backend, docker-compose, and frontend.
- appsettings.json Editor:** The current file being edited. It contains configuration settings for logging, allowed hosts, and connection strings. The connection string for MariaDB is highlighted in blue.
- Status Bar:** 100%, Aucun problème détecté, Lig. : 9, Car. : 1, SPC, CRLF.
- Bottom Navigation:** Explorateur de solutions, Modifications Git.

```
1  {
2    "Logging": {
3      "LogLevel": {
4        "Default": "Information",
5        "Microsoft.AspNetCore": "Warning"
6      },
7    },
8    "AllowedHosts": "*",
9    "ConnectionStrings": {
10      "MariaDbConnectionString": "Server=database;User ID=pifou;Password=pasglop;Port=3306;Database=db"
11    }
12 }
```

The screenshot shows the Visual Studio IDE interface with the following details:

- Top Bar:** Fichier, Edition, Affichage, Git, Projet, Générer, Déboguer, Test, Analyser, Outils, Extensions, Fenêtre, Aide, Rechercher (Ctrl+Q), account-manager, Connexion.
- Toolbar:** Includes icons for Undo, Redo, Save, Build, Run, and Docker.
- Project Explorer:** Shows three projects: backend, docker-compose, and frontend.
- Solution Explorer:** Displays the file structure for each project, including files like Dockerfile, appsettings.json, and various configuration files.
- Code Editor:** The main window shows the `Program.cs` file for the `backend` project, which configures an ASP.NET Core application with Swagger and Docker support.
- Status Bar:** Shows "100 %", "Aucun problème détecté", and other build-related status.
- Output Window:** Shows logs related to Docker Desktop, indicating it's unable to determine images to pull and suggesting a restart.

```
1 var builder = WebApplication.CreateBuilder(args);
2
3 // Add services to the container.
4
5 builder.Services.AddControllers();
6 // Learn more about configuring Swagger/OpenAPI at https://aka.ms/aspnetcore/swashbuckle
7 builder.Services.AddEndpointsApiExplorer();
8 builder.Services.AddSwaggerGen();
9
10 var app = builder.Build();
11
12 // Configure the HTTP request pipeline.
13 if (app.Environment.IsDevelopment()) {
14     app.UseSwagger();
15     app.UseSwaggerUI();
16 }
17
18 app.UseAuthorization();
19
20 app.MapControllers();
21
22 app.Run();
23
```

Autorisation : 100 % Aucun problème détecté | Sortie | Afficher la sortie à partir de : Outils de conteneur | Impossible de déterminer les images à tirer (pull).  
Démarrage de Docker Desktop...  
Docker Desktop ne fonctionne pas comme prévu. Essayez de redémarrer Docker.

Rechercher dans Explorateur de solutions (Ctrl+\$)

Explorateur de solutions | Modifications Git

## Program.cs

```
3  using Pomelo.EntityFrameworkCore.MySql.Infrastructure;
4
5  var builder = WebApplication.CreateBuilder(args);
6
7  // Add services to the container.
8  builder.Services.AddControllers();
9  var connectionString = builder.Configuration.GetConnectionString("MariaDbConnectionString");
10 builder.Services.AddDbContext<DatabaseContext>(options => options.UseMySQL(
11     connectionString,
12     ServerVersion.AutoDetect(connectionString)
13 ));
14 builder.Services.AddControllers();
15 builder.Services.AddEndpointsApiExplorer();
16 builder.Services.AddSwaggerGen();
17
18 var app = builder.Build();
19
20 // Configure the HTTP request pipeline.
21 if (!app.Environment.IsDevelopment()) {
22     app.UseDefaultFiles();
23     app.UseStaticFiles();
24 }
25
26 app.UseAuthorization();
27 app.MapControllers();
28
29 app.UseSwagger();
30 app.UseSwaggerUI();
31
32 app.Run();
```

## Explorateur de solutions

Rechercher dans Explorateur de solutions (Ctrl+\$)

Solution 'account-manager' (3 sur 3 de projets)

- backend
  - Connected Services
  - Dépendances
  - Properties
  - Controllers
  - DTOs
  - Models
  - Services
  - appsettings.json
  - Dockerfile
- C# Program.cs

## docker-compose

- frontend
  - npm
  - public
  - src
    - .gitignore
    - aspnetcore-https.js
    - aspnetcore-react.js
    - Dockerfile
    - nuget.config
    - package.json
    - package-lock.json
    - README.md

# 8

*Set backend classes*

- Nouvel élément... Ctrl+Maj+A
- Élément existant... Maj+Alt+A
- Nouvel élément généré automatiquement...
- Nouveau dossier
- Prise en charge de Docker...
- Application Insights Telemetry...
- Bibliothèque côté client...
- Modèle de Machine Learning...
- Nouveau projet de tâche web Azure
- Projet existant en tant que tâche web Azure
- Référence de projet...
- Référence de projet partagé...
- Référence COM...
- Référence de service...
- Service connecté
- Classe...
- New EditorConfig
- Nouvel EditorConfig (IntelliCode)

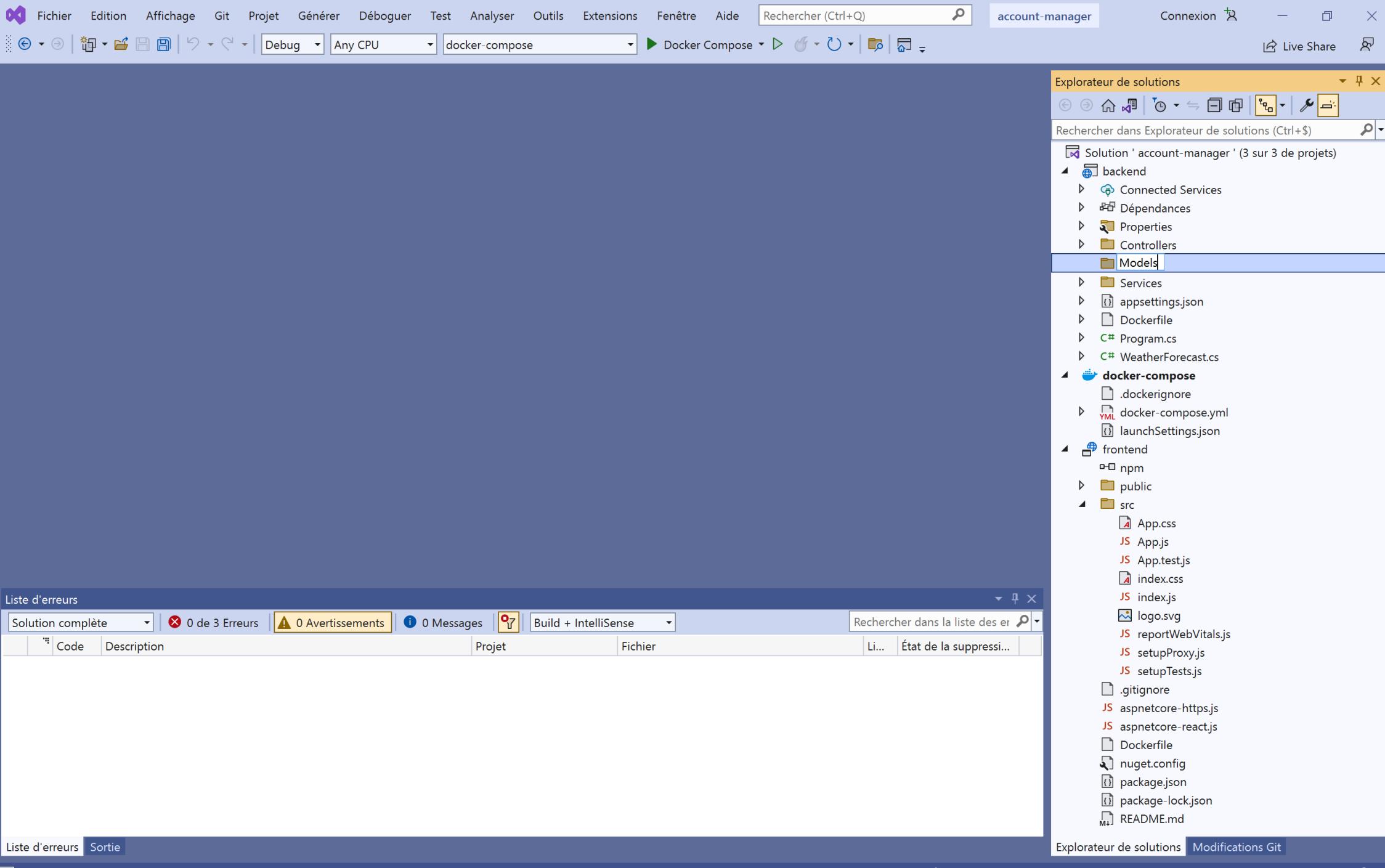
- Générer
- Regénérer
- Nettoyer
- Affichage
- Analyser et nettoyer le code
- Compresser
- Publier...
- Configurer Application Insights...
- Vue d'ensemble
- Limiter à ceci
- Nouvelle vue Explorateur de solutions
- Imbrication de fichiers
- Modifier le fichier projet
- Dépendances de build
- Ajouter**
- Gérer les packages NuGet...
- Gérer les bibliothèques côté client...
- Gérer les données secrètes de l'utilisateur
- Supprimer les références inutilisées...
- Synchroniser les espaces de noms
- Définir en tant que projet de démarrage
- Déboguer
- Couper Ctrl+X
- Supprimer Suppr
- Renommer F2
- Décharger le projet
- Charger les dépendances directes
- Charger l'ensemble de l'arborescence des dépendances
- Copier le chemin d'accès complet
- Ouvrir le dossier dans l'Explorateur de fichiers
- Ouvrir dans Terminal
- Propriétés Alt+Entrée

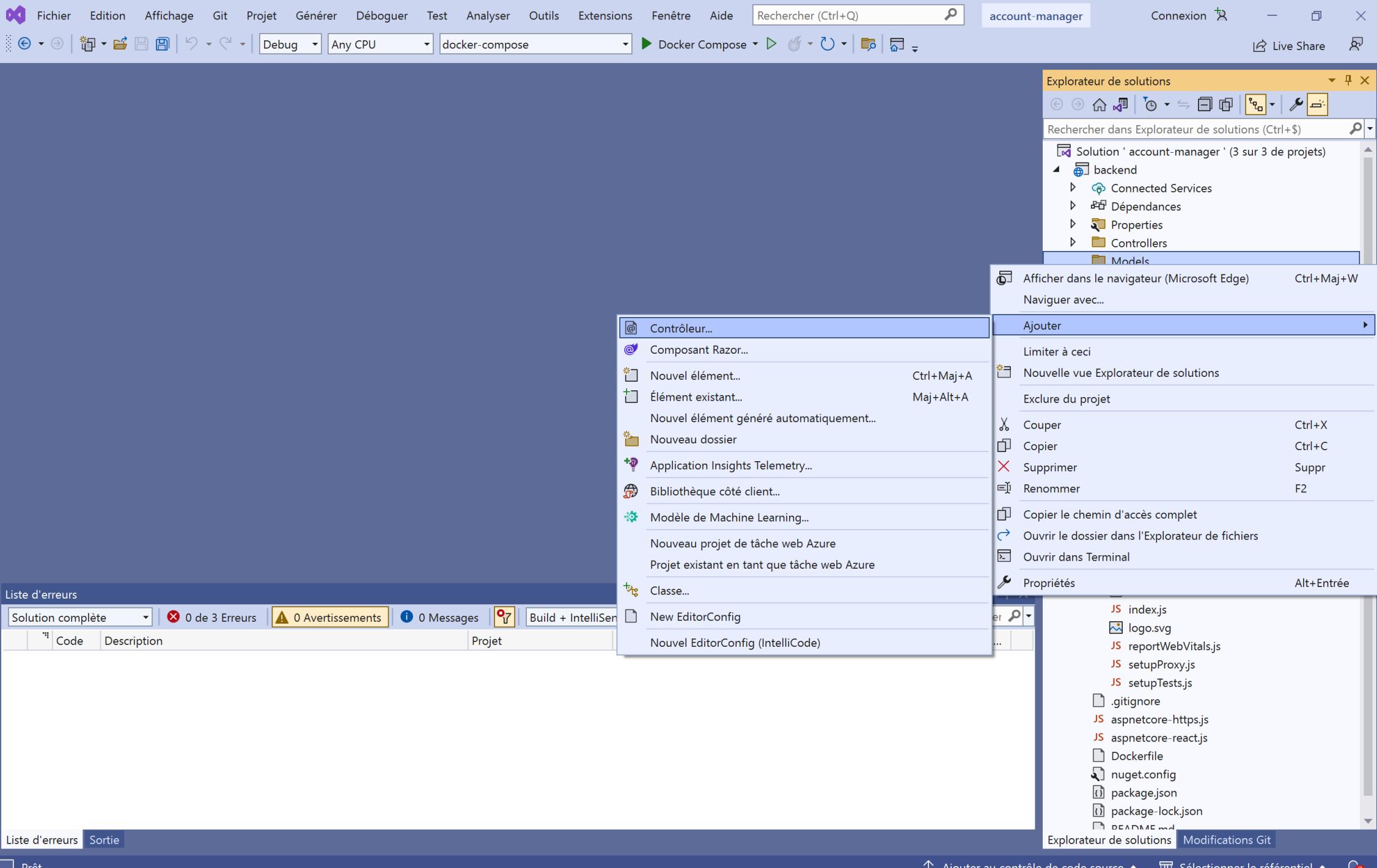
## Liste d'erreurs

Solution complète		0 de 3 Erreurs	0 Avertissements	0 Messages	Build + IntelliSense	Rechercher dans la
#	Code	Description	Projet	Fichier	Li...	État de la

Liste d'erreurs Sortie

Prêt





## Ajouter un nouvel élément - backend

? X

## Installé

Trier par : Par défaut



Rechercher (Ctrl+E)

## C#

- Général
- ASP.NET Core

## En ligne

	Classe	C#
	Interface	C#
	Composant Razor	C#
	Contrôleur MVC avec des actions de lecture/écriture	C#
	Contrôleur MVC - Vide	C#
	Contrôleur d'API - Vide	C#
	Contrôleur d'API avec actions de lecture/écriture	C#
	Page Razor - Vide	C#
	Vue Razor - Vide	C#
	Disposition Razor	C#
	Début de la vue Razor	C#
	Fichier de code	C#
	Fichier d'informations de l'assembly	C#
	Machine Learning Model (ML.NET)	C#

Type : C#

Déclaration de classe vide

Nom :

User.cs

Ajouter

Annuler

The screenshot shows a Visual Studio interface with the following details:

- Top Bar:** Fichier, Edition, Affichage, Git, Projet, Générer, Déboguer, Test, Analyser, Outils, Extensions, Fenêtre, Aide, Rechercher (Ctrl+Q), account-manager, Connexion.
- Toolbar:** Includes icons for file operations like Open, Save, Print, and a Docker Compose icon.
- Project Selector:** Debug, Any CPU, docker-compose.
- Solution Explorer:** Explorateur de solutions pane showing the project structure:
  - Solution 'account-manager' (3 sur 3 de projets)
    - backend
      - Connected Services
      - Dépendances
      - Properties
      - Controllers
      - Models
        - User.cs
    - docker-compose
      - .dockerignore
      - docker-compose.yml
      - launchSettings.json
    - frontend
      - npm
      - public
      - src
        - App.css
        - App.js
        - App.test.js
        - index.css
        - index.js
        - logo.svg
        - reportWebVitals.js
        - setupProxy.js
        - setupTests.js
- Code Editor:** User.cs file in the backend project, showing the following code:

```
namespace backend.Models {  
    public class User {  
    }  
}
```
- Toolbars:** Standard toolbar at the bottom.
- Status Bar:** 100%, Aucun problème détecté, Lig. : 5, Car. : 1, SPC, CRLF.
- Bottom Navigation:** Liste d'erreurs..., Sortie, Explorateur de solutions, Modifications Git.

The screenshot shows a Visual Studio interface with the following details:

- Top Bar:** Fichier, Edition, Affichage, Git, Projet, Générer, Déboguer, Test, Analyser, Outils, Extensions, Fenêtre, Aide, Rechercher (Ctrl+Q), account-manager, Connexion.
- Toolbar:** Includes icons for file operations like Open, Save, Print, and a Docker Compose icon.
- Project Selector:** Debug, Any CPU, docker-compose.
- Main Area:** A code editor window titled "User.cs\*" showing the following C# code:

```
namespace backend.Models {    public class User {        public int Id { get; set; }        public string? Name { get; set; }        public string? Email { get; set; }        public string? Password { get; set; }    }}
```
- Solution Explorer:** Explorateur de solutions showing the project structure:
  - Solution 'account-manager' (3 sur 3 de projets)
    - backend
      - Connected Services
      - Dépendances
      - Properties
      - Controllers
      - Models
        - User.cs
      - Services
        - DatabaseContext.cs
      - appsettings.json
      - Dockerfile
      - Program.cs
      - WeatherForecast.cs
    - docker-compose
      - .dockerignore
      - docker-compose.yml
      - launchSettings.json
    - frontend
      - npm
      - public
      - src
        - App.css
        - App.js
        - App.test.js
        - index.css
        - index.js
        - logo.svg
        - reportWebVitals.js
        - setupProxy.js
        - setupTests.js
- Error List:** Liste d'erreurs... showing 0 errors, 0 warnings, and 0 messages.
- Bottom Status Bar:** Prêt, Ajouter au contrôle de code source, Sélectionner le référentiel, Modifications Git.

- Nouvel élément... Ctrl+Maj+A
- Élément existant... Maj+Alt+A
- Nouvel élément généré automatiquement...
- Nouveau dossier
- Prise en charge de Docker...
- Application Insights Telemetry...
- Bibliothèque côté client...
- Modèle de Machine Learning...
- Nouveau projet de tâche web Azure
- Projet existant en tant que tâche web Azure
- Référence de projet...
- Référence de projet partagé...
- Référence COM...
- Référence de service...
- Service connecté
- Classe...
- New EditorConfig
- Nouvel EditorConfig (IntelliCode)

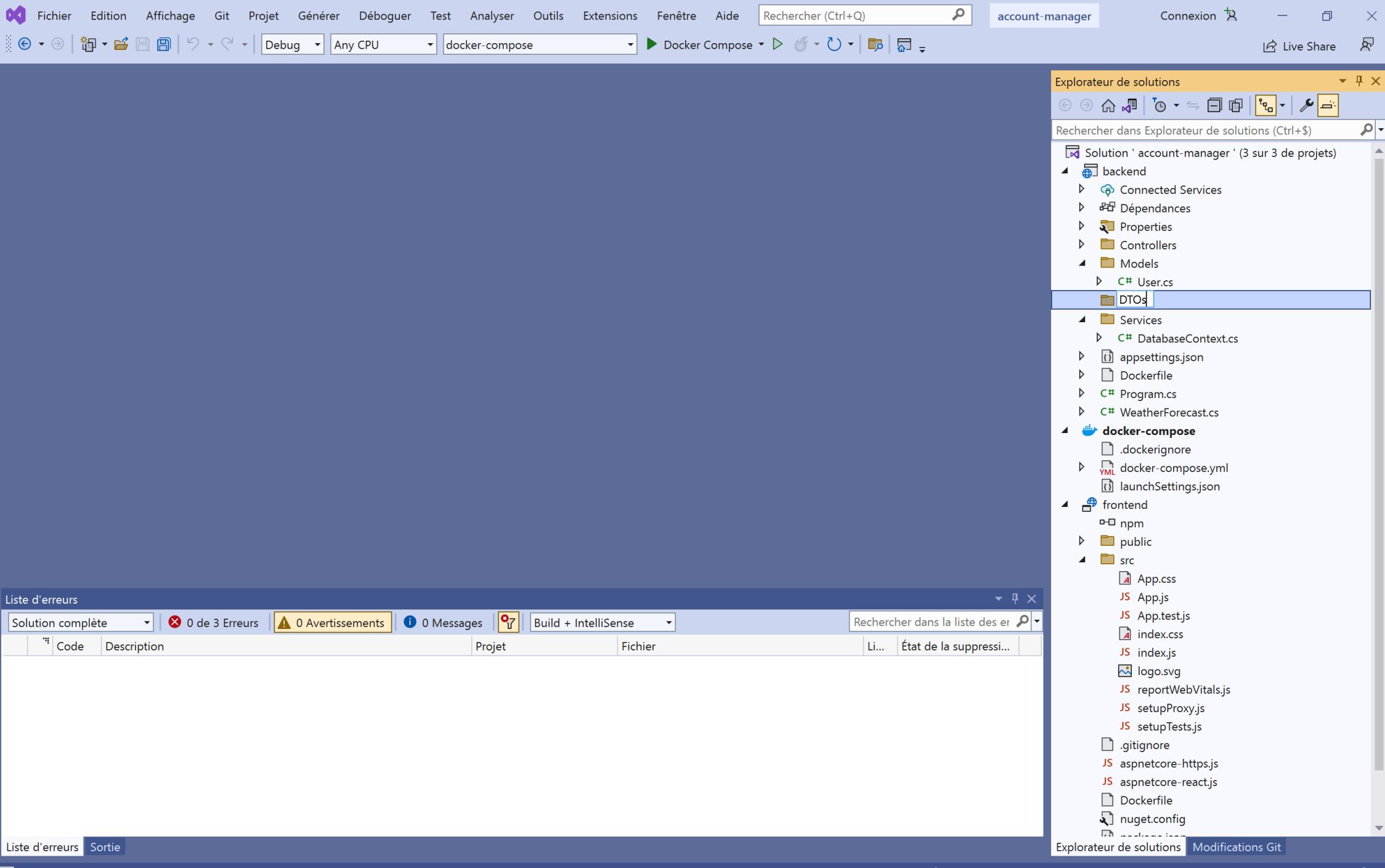
- Générer
- Regénérer
- Nettoyer
- Affichage
- Analyser et nettoyer le code
- Compresser
- Publier...
- Configurer Application Insights...
- Vue d'ensemble
- Limiter à ceci
- Nouvelle vue Explorateur de solutions
- Imbrication de fichiers
- Modifier le fichier projet
- Dépendances de build
- Ajouter**
- Gérer les packages NuGet...
- Gérer les bibliothèques côté client...
- Gérer les données secrètes de l'utilisateur
- Supprimer les références inutilisées...
- Synchroniser les espaces de noms
- Définir en tant que projet de démarrage
- Déboguer
- Couper Ctrl+X
- Supprimer Suppr
- Renommer F2
- Décharger le projet
- Charger les dépendances directes
- Charger l'ensemble de l'arborescence des dépendances
- Copier le chemin d'accès complet
- Ouvrir le dossier dans l'Explorateur de fichiers
- Ouvrir dans Terminal
- Propriétés Alt+Entrée

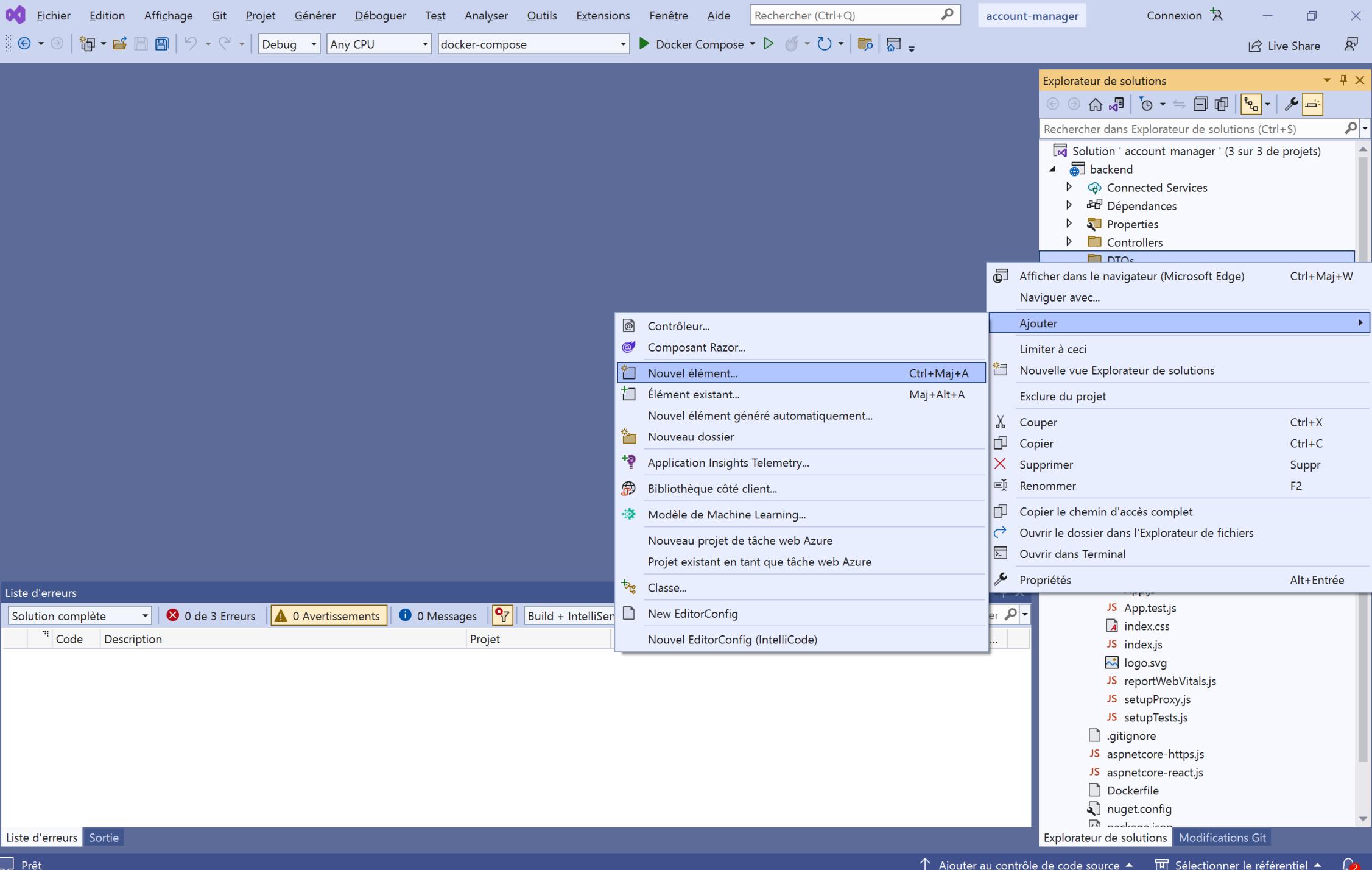
## Liste d'erreurs

Solution complète		0 de 3 Erreurs	0 Avertissements	0 Messages	Build + IntelliSense	Rechercher dans la
#	Code	Description	Projet	Fichier	Li...	État de la

Liste d'erreurs Sortie

Prêt





## Ajouter un nouvel élément - backend

? X

## Installé

Trier par : Par défaut



Rechercher (Ctrl+E)

## C#

- Général
- ASP.NET Core

## En ligne

	Classe	C#
	Interface	C#
	Composant Razor	C#
	Contrôleur MVC avec des actions de lecture/écriture	C#
	Contrôleur MVC - Vide	C#
	Contrôleur d'API - Vide	C#
	Contrôleur d'API avec actions de lecture/écriture	C#
	Page Razor - Vide	C#
	Vue Razor - Vide	C#
	Disposition Razor	C#
	Début de la vue Razor	C#
	Fichier de code	C#
	Fichier d'informations de l'assembly	C#
	Machine Learning Model (ML.NET)	C#

Type : C#

Déclaration de classe vide

Nom :

UserDto.cs

Ajouter

Annuler

The screenshot shows a Visual Studio interface with the following details:

- Top Bar:** Fichier, Edition, Affichage, Git, Projet, Générer, Déboguer, Test, Analyser, Outils, Extensions, Fenêtre, Aide, Rechercher (Ctrl+Q), account-manager, Connexion.
- Toolbar:** Standard icons for file operations like Open, Save, Print, etc.
- Project Explorer:** Shows three projects: backend, docker-compose, and frontend. The backend project is expanded, showing its structure: Connected Services, Dépendances, Properties, Controllers, DTOs (containing UserDto.cs), Models, User.cs, Services, DatabaseContext.cs, appsettings.json, Dockerfile, Program.cs, WeatherForecast.cs.
- Code Editor:** The UserDto.cs file is open in the editor. The code is as follows:

```
1 namespace backend.DTOs {  
2     public class UserDto {  
3     }  
4 }  
5
```

- Solution Explorer:** Shows the solution structure with three projects: backend, docker-compose, and frontend.
- Toolbars:** Standard Visual Studio toolbars for navigation and selection.
- Status Bar:** 100%, Aucun problème détecté, Lig. : 1, Car. : 1, SPC, CRLF.
- Bottom Navigation:** Liste d'erreurs..., Sortie, Explorateur de solutions, Modifications Git.

UserDto.cs\*

```
backend
└── backend.DTOS.UserPostDto
    └── Toltem(UserPostDto dto)

1  using backend.Models;
2
3  namespace backend.DTOS {
4
5      2 références
6      public class UserGetDto {
7
8          1 référence
9          public int Id { get; set; }
10
11         1 référence
12         public string? Name { get; set; }
13
14         1 référence
15         public string? Email { get; set; }
16
17         0 références
18         public static UserGetDto ToDto(User item) {
19             return new UserGetDto {
20                 Id = item.Id,
21                 Name = item.Name,
22                 Email = item.Email,
23             };
24         }
25
26         1 référence
27         public class UserPostDto {
28
29             1 référence
30             public string? Name { get; set; }
31
32             1 référence
33             public string? Email { get; set; }
34
35             1 référence
36             public string? Password { get; set; }
37
38             0 références
39             public static User ToItem(UserPostDto dto) {
40                 return new User {
41                     Name = dto.Name,
42                     Email = dto.Email,
43                     Password = dto.Password,
44                 };
45             }
46         }
47     }
48 }
```

Explorateur de solutions

Rechercher dans Explorateur de solutions (Ctrl+\$)

Solution 'account-manager' (3 sur 3 de projets)

- backend
  - Connected Services
  - Dépendances
  - Properties
  - Controllers
  - DTOS
    - C# UserDto.cs
    - Models
      - C# User.cs
- docker-compose
  - .dockerrcignore
  - docker-compose.yml
  - launchSettings.json
- frontend
  - npm
  - public
  - src
    - App.css
    - App.js
    - App.test.js
    - index.css
    - index.js
    - logo.svg
    - reportWebVitals.js
    - setupProxy.js
    - setupTests.js
    - .gitignore
    - aspnetcore-https.js
    - aspnetcore-react.js
    - Dockerfile
    - nuget.config

Explorateur de solutions

Rechercher dans Explorateur de solutions (Ctrl+\$)

- Solution 'account-manager' (3 sur 3 de projets)
  - backend
    - Connected Services
    - Dépendances
    - Properties
  - Controllers

Afficher dans le navigateur (Microsoft Edge) Ctrl+Maj+W

Naviguer avec...

Ajouter

Limiter à ceci

- Nouvelle vue Explorateur de solutions
- Exclure du projet
- Couper Ctrl+X
- Copier Ctrl+C
- Supprimer Suppr
- Renommer F2
- Copier le chemin d'accès complet
- Ouvrir le dossier dans l'Explorateur de fichiers
- Ouvrir dans Terminal

Propriétés Alt+Entrée

public

- src
  - App.css
  - App.js
  - App.test.js
  - index.css
  - index.js
  - logo.svg
  - reportWebVitals.js
  - setupProxy.js
  - setupTests.js
- .gitignore
- aspnetcore-https.js
- aspnetcore-react.js

Explorateur de solutions Modifications Git



X

## Ajouter un nouvel élément généré automatiquement

### Installé

#### Éléments communs

API

Composant Razor

#### MVC

Affichage

Contrôleur

Pages Razor

Identité

Mise en page



Contrôleur MVC avec des actions de lecture/écriture



Contrôleur MVC avec vues, utilisant Entity Framework



Contrôleur MVC - Vide

### Contrôleur MVC avec des actions de lecture/écriture

par Microsoft

v1.0.0.0

Contrôleur MVC avec des actions permettant de créer, lire, mettre à jour, supprimer et répertorier des entités.

ID : MvcControllerWithActionsScaffolder

Ajouter

Annuler

## Ajouter un nouvel élément - backend

? X

## Installé

Trier par : Par défaut



Rechercher (Ctrl+E)

## C#

Général

ASP.NET Core

## En ligne

- |    |   |    |
|----|---|----|
| C# | Classe  | C# |
| C# | Interface   | C# |
| C# | Composant Razor                                     | C# |
| C# | Contrôleur MVC avec des actions de lecture/écriture | C# |
| C# | Contrôleur MVC - Vide                               | C# |
| C# | Contrôleur d'API - Vide                             | C# |
| C# | Contrôleur d'API avec actions de lecture/écriture   | C# |
| C# | Page Razor - Vide                                   | C# |
| C# | Vue Razor - Vide                                    | C# |
| C# | Disposition Razor                                   | C# |
| C# | Début de la vue Razor                               | C# |
| C# | Fichier de code                                     | C# |
| C# | Fichier d'informations de l'assembly                | C# |
| C# | Machine Learning Model (ML.NET)                     | C# |

Type : C#

Contrôleur d'application MVC avec des actions permettant de créer, mettre à jour, supprimer et lister des entités

Nom :

UserController.cs

Ajouter

Annuler

The screenshot shows a Microsoft Visual Studio interface with the following details:

- Top Bar:** Fichier, Edition, Affichage, Git, Projet, Générer, Déboguer, Test, Analyser, Outils, Extensions, Fenêtre, Aide, Rechercher (Ctrl+Q), account-manager, Connexion.
- Toolbar:** Includes icons for Undo, Redo, Save, Open, and various project-related functions.
- Code Editor:** The main window displays the file `UserController.cs` from the `backend` project. The code implements a `UserController` with methods for Index, Details, Create, and Edit actions. It uses `Microsoft.AspNetCore.Http` and `Microsoft.AspNetCore.Mvc` namespaces.
- Solution Explorer:** Shows the solution structure with three projects:
  - `backend`: Contains Connected Services, Dependencies, Properties, and Controllers (which contains `UserController.cs` and `WeatherForecastController.cs`).
  - `docker-compose`: Contains `.dockerrcignore`, `docker-compose.yml`, and `launchSettings.json`.
  - `frontend`: Contains `npm`, `public`, and `src` folder with files like `App.css`, `App.js`, `App.test.js`, `index.css`, `index.js`, and `logo.svg`.
- Status Bar:** Includes the percentage (91%), a warning icon, the message "Aucun problème détecté" (No problems detected), and file navigation information (Lig. : 34, Car. : 27, SPC, CRLF).
- Bottom Bar:** Includes "Prêt" (Ready), "Ajouter au contrôle de code source" (Add to code control), "Sélectionner le référentiel" (Select the reference), and a Git icon.

```
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using backend.DTOs;
using backend.Models;
using backend.Services;

namespace backend.Controllers {

    [Route("api/users")]
    [ApiController]
    public class UserController : ControllerBase {

        private readonly DatabaseContext _context;

        public UserController(DatabaseContext context) { _context = context; }

        // ...

        private bool Exists(long id) {
            return _context.Users.Any(e => e.Id == id);
        }
    }
}
```

```
// GET: api/users
[HttpGet]
public async
Task<ActionResult<IEnumerable<UserGetDto>>>
GetAll() {
    return await _context.Users
        .Select(x =>
UserGetDto.ToDto(x))
        .ToListAsync();
}

// GET: api/users/5
[HttpGet("{id}")]
public async
Task<ActionResult<UserGetDto>> Get(long id)
{
    var item = await
_context.Users.FindAsync(id);
    if (item == null) return
NotFound();

    return UserGetDto.ToDto(item);
}
```

```
// PUT: api/users/5
    [HttpPut("{id}")]
    public async Task<IActionResult>
Put(long id, UserPostDto dto) {
    var item = await
_context.Users.FindAsync(id);
    if (item == null) return
NotFound();

    item.Name = dto.Name;
    item.Email = dto.Email;
    item.Password = dto.Password;

    try {
        await
_context.SaveChangesAsync();
    } catch
(DbUpdateConcurrencyException) when
(!Exists(id)) {
        return NotFound();
    }

    return NoContent();
}
```

```
// POST: api/users
    [HttpPost]
    public async
Task<ActionResult<UserGetDto>>
Post(UserPostDto dto) {
    var item =
UserPostDto.ToItem(dto);

    _context.Users.Add(item);
    await
_context.SaveChangesAsync();

    return CreatedAtAction(
        nameof(Get),
        new { id = item.Id },
        UserGetDto.Dto(item)
    );
}
```

```
// DELETE: api/users/5
[HttpDelete("{id}")]
public async Task<IActionResult>
Delete(long id) {
    var item = await
_context.Users.FindAsync(id);
    if (item == null) return
NotFound();

    _context.Users.Remove(item);
    await
_context.SaveChangesAsync();

    return NoContent();
}
```

Eichier Edition Affichage Git Projet Générer Déboguer Test Analyser Outils Extensions Fenêtre Aide Rechercher (Ctrl+Q) account-manager Connexion Live Share

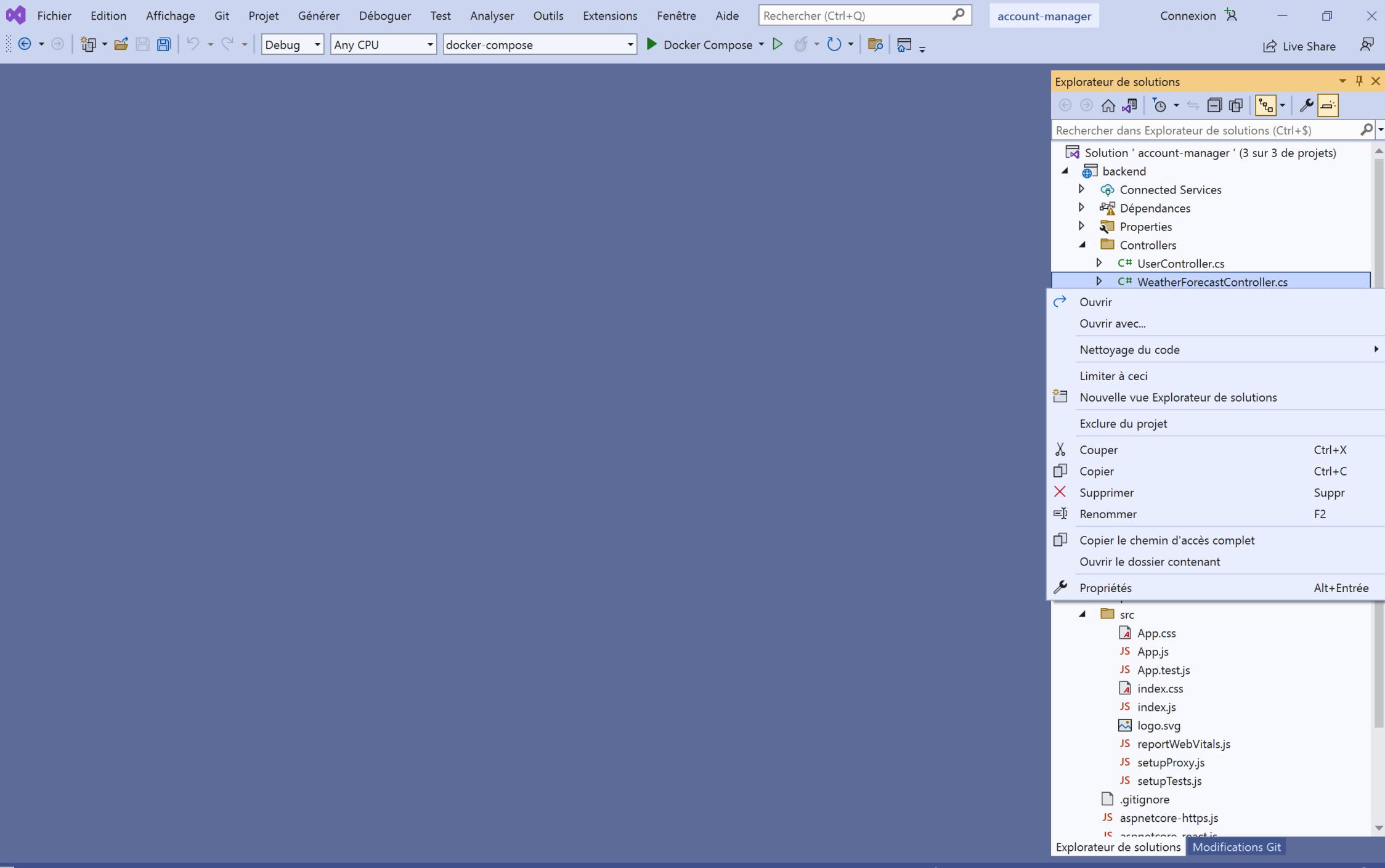
Backend UserController.cs

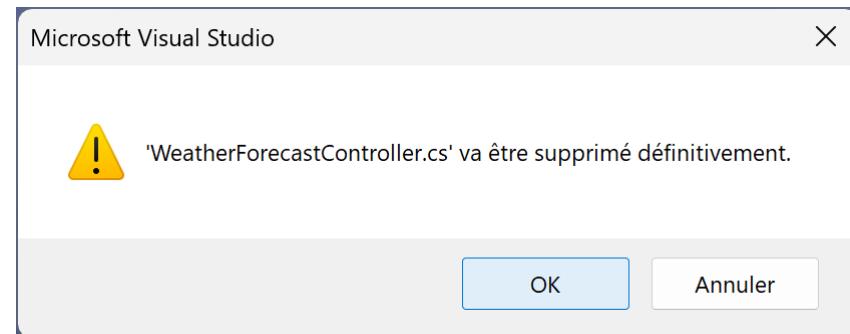
```
1 using Microsoft.AspNetCore.Mvc;
2 using Microsoft.EntityFrameworkCore;
3 using backend.DTOs;
4 using backend.Models;
5 using backend.Services;
6
7 namespace backend.Controllers {
8
9     [Route("api/users")]
10    [ApiController]
11    public class UserController : ControllerBase {
12
13        private readonly DatabaseContext _context;
14
15        public UserController(DatabaseContext context) { _context = context; }
16
17        // GET: api/users
18        [HttpGet]
19        public async Task<ActionResult<IEnumerable<UserGetDto>>> GetAll() {
20            return await _context.Users
21                .Select(x => UserGetDto.ToDto(x))
22                .ToListAsync();
23        }
24
25        // GET: api/users/5
26        [HttpGet("{id}")]
27        public async Task<ActionResult<UserGetDto>> Get(long id) {
28            var item = await _context.Users.FindAsync(id);
29            if (item == null) return NotFound();
30
31            return UserGetDto.ToDto(item);
32        }
33
34        // PUT: api/users/5
35        [HttpPut("{id}")]
36        public async Task<IActionResult> Put(long id, UserPostDto dto) {
37            var item = await _context.Users.FindAsync(id);
38            if (item == null) return NotFound();
39
40            item.Name = dto.Name;
41            item.Email = dto.Email;
42            item.Password = dto.Password;
43
44            try {
45                await _context.SaveChangesAsync();
46            } catch (DbUpdateConcurrencyException) {
47                if (!UserExists(id)) return NotFound();
48            }
49
50            return NoContent();
51        }
52
53        // POST: api/users
54        [HttpPost]
55        public async Task<IActionResult> Post(UserPostDto dto) {
56            var user = new User();
57            user.Name = dto.Name;
58            user.Email = dto.Email;
59            user.Password = dto.Password;
60
61            _context.Users.Add(user);
62            await _context.SaveChangesAsync();
63
64            return CreatedAtAction("Get", new { id = user.Id }, user);
65        }
66
67        // DELETE: api/users/5
68        [HttpDelete("{id}")]
69        public async Task<IActionResult> Delete(long id) {
70            var item = await _context.Users.FindAsync(id);
71            if (item == null) return NotFound();
72
73            _context.Users.Remove(item);
74            await _context.SaveChangesAsync();
75
76            return NoContent();
77        }
78    }
79}
```

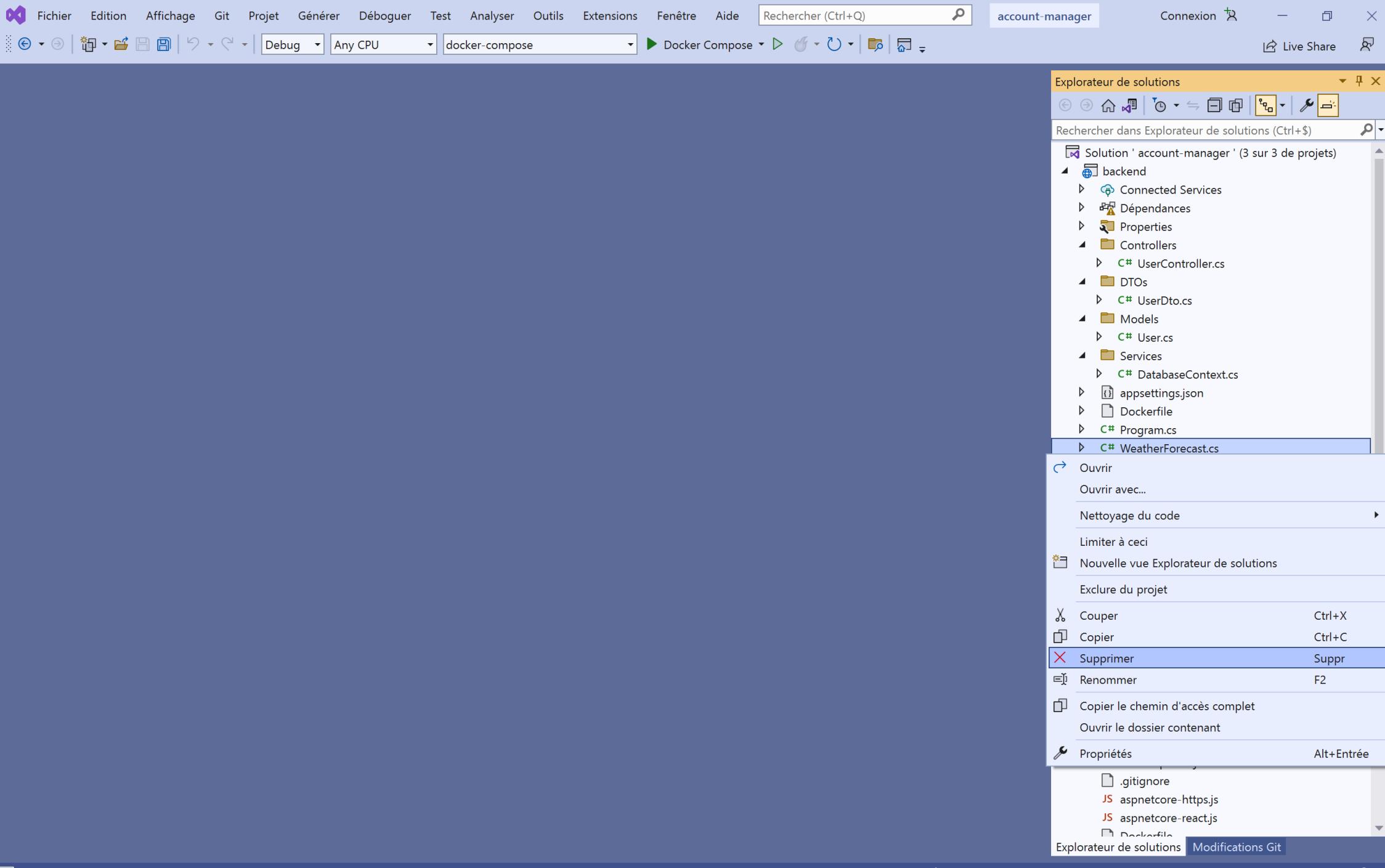
Explorateur de solutions

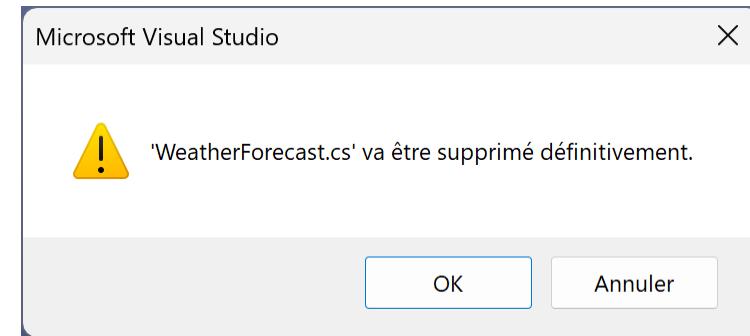
Solution 'account-manager' (3 sur 3 de projets)

- backend
  - Connected Services
  - Dépendances
  - Properties
  - Controllers
    - C# UserController.cs
    - C# WeatherForecastController.cs
  - DTOs
    - C# UserDto.cs
  - Models
    - C# User.cs
  - Services
    - C# DatabaseContext.cs
  - appsettings.json
  - Dockerfile
  - Program.cs
  - C# WeatherForecast.cs
- docker-compose
  - .dockerrignore
  - docker-compose.yml
  - launchSettings.json
- frontend
  - npm
  - public
  - src
    - App.css
    - App.js
    - App.test.js
    - index.css
    - index.js
    - logo.svg
    - reportWebVitals.js
    - setupProxy.js
    - setupTests.js
  - .gitignore
  - aspnetcore-https.js









9

Add *reverse-proxy* instance

account-manager

Nouveau | Trier | Afficher | ...

Kevin Doolaege > source > repos > account-manager

Rechercher dans : account-manager

Accueil

OneDrive - Personal

Ce PC

NAS605B22

Local Disk (C:)

Bureau

Téléchargements

Documents

Images

Kevin

Ce PC

Dossier personnel (NAS605B2)

Multimédia (NAS605B22)

Local Disk (C:)

Bibliothèques

Réseau

Linux

Nom | Modifié le | Type | Taille

Nom	Modifié le	Type	Taille
backend	11/02/2023 17:54	Dossier de fichiers	
database	11/02/2023 17:55	Dossier de fichiers	
frontend	11/02/2023 17:50	Dossier de fichiers	
obj	11/02/2023 17:54	Dossier de fichiers	
reverse-proxy	11/02/2023 17:55	Dossier de fichiers	
.dockerignore	11/02/2023 17:54	Fichier DOCKERIGNO...	1 Ko
account-manager.sln	11/02/2023 17:54	Visual Studio Solution	3 Ko
docker-compose.dcproj	11/02/2023 17:54	Fichier DCPROJ	1 Ko
docker-compose.override.yml	11/02/2023 17:54	Fichier YML	1 Ko
docker-compose.yml	11/02/2023 17:54	Fichier YML	1 Ko

10 élément(s) |

reverse-proxy

Nouveau | X | +

Trier | Afficher | ...

← → ↑ ▶ Kevin Doolaege > source > repos > account-manager > reverse-proxy

Rechercher dans : reverse-proxy

Accueil

OneDrive - Personal

Ce PC

NAS605B22

Local Disk (C:)

Bureau

Téléchargements

Documents

Images

Kevin

Ce PC

Dossier personnel (NAS605B2)

Multimédia (NAS605B22)

Local Disk (C:)

Bibliothèques

Réseau

Linux

2 élément(s) |

Nom	Modifié le	Type	Taille
Dockerfile	11/02/2023 19:47	Fichier	0 Ko
nginx.conf	11/02/2023 19:47	Fichier CONF	0 Ko



Fichier Modifier Affichage

```
FROM nginx:alpine
COPY reverse-proxy/nginx.conf /etc/nginx/nginx.conf
RUN rm /etc/nginx/conf.d/default.conf

RUN apk update && apk add openssl
RUN openssl req -x509 -nodes -days 1461 \
    -subj "/C=FR/ST=Lille/O=SM/CN=account-manager.site" \
    -addext "subjectAltName=DNS:account-manager.site" -newkey rsa:2048 \
    -keyout /etc/ssl/private/app-cert.key -out /etc/ssl/certs/app-cert.crt

EXPOSE 80 443
ENTRYPOINT [ "nginx", "-g", "daemon off;" ]
```

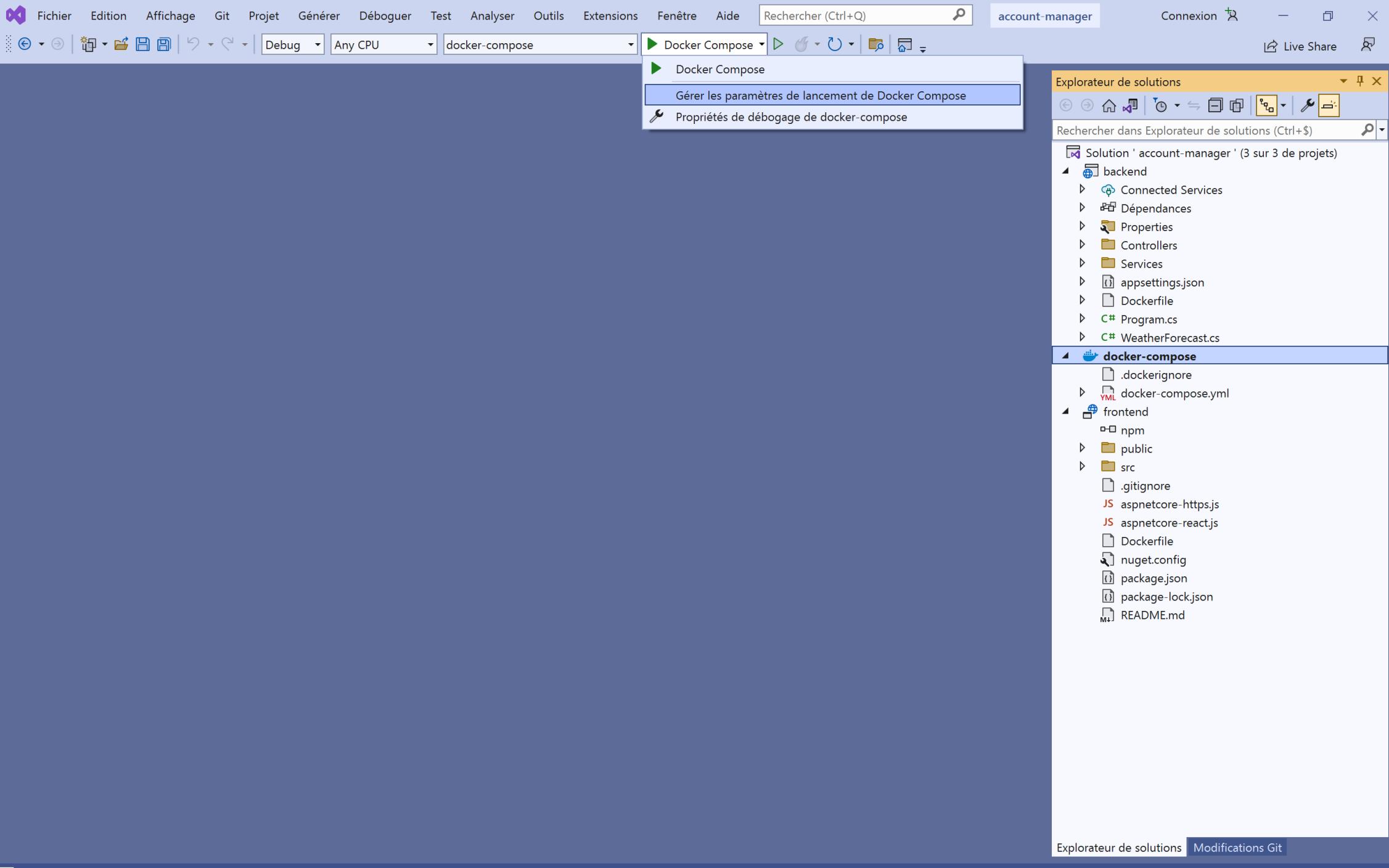
\*nginx.conf - Bloc-notes

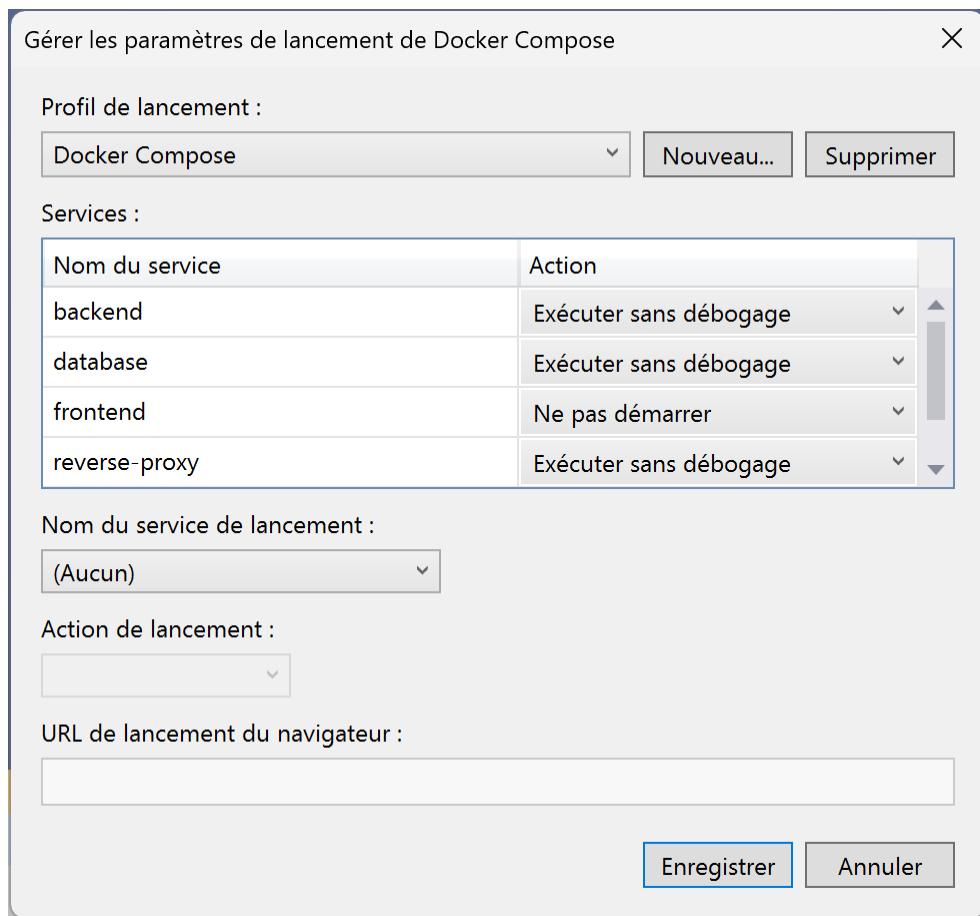
Fichier    Modifier    Affichage

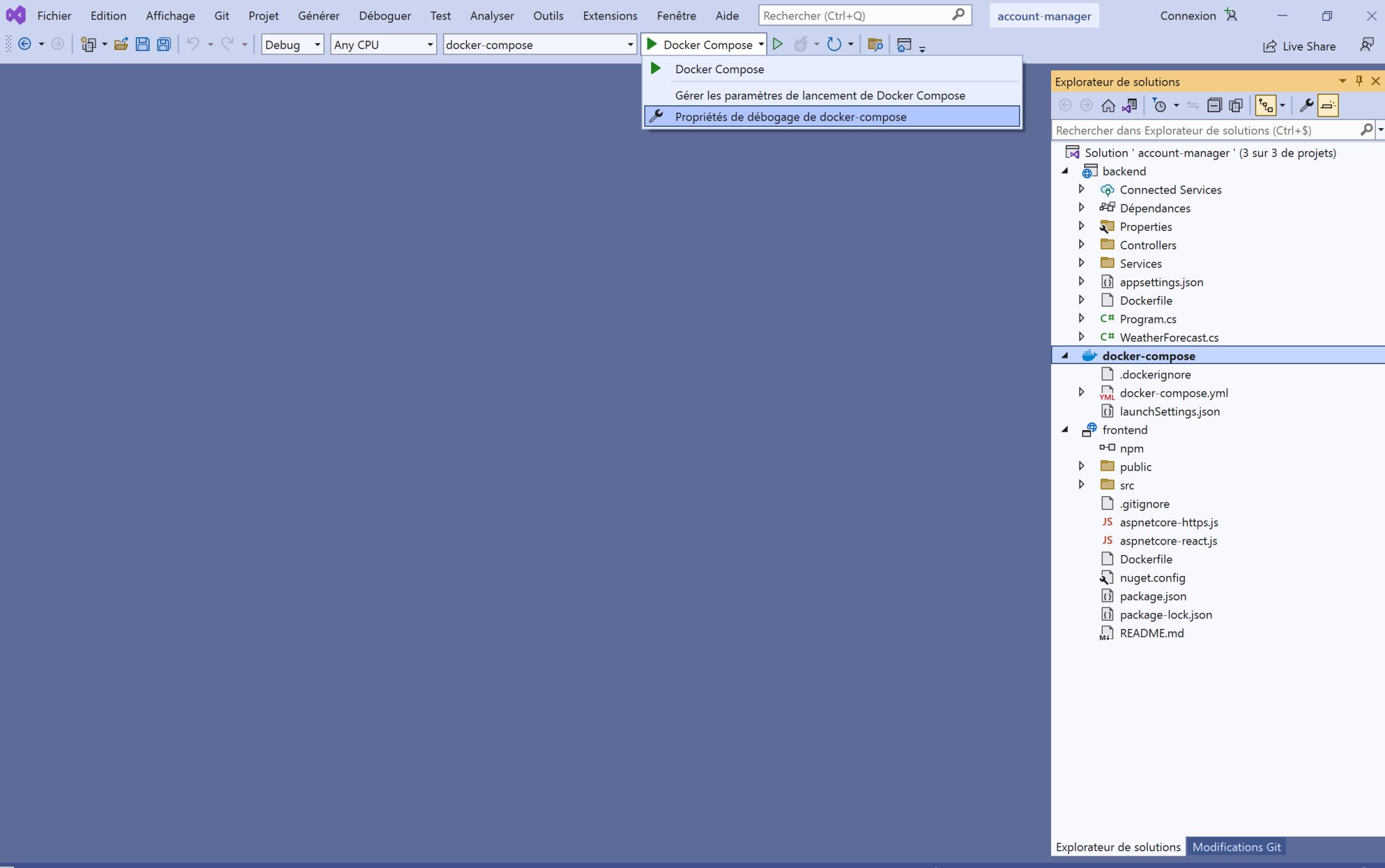
events {  
}  
  
http {  
    server {  
        listen 80;  
        listen [::]:80;  
  
        return 301 https://\$host\$request\_uri;  
    }  
  
    server {  
        listen 443 ssl default\_server;  
        listen [::]:443 ssl http2 default\_server;  
  
        ssl\_certificate /etc/ssl/certs/app-cert.crt;  
        ssl\_certificate\_key /etc/ssl/private/app-cert.key;  
  
        access\_log /var/log/nginx/access.log;  
        error\_log /var/log/nginx/error.log;  
  
        error\_page 500 502 503 504 /50x.html;  
        location = /50x.html {  
            root /usr/share/nginx/html;  
        }  
  
        location / {  
            proxy\_pass http://frontend/;  
        }  
  
        location /swagger {  
            proxy\_pass http://backend/swagger;  
        }  
  
        location /api {  
            proxy\_pass http://backend/api;  
        }  
    }  
}

10

*Set docker-compose settings*







Pages de propriétés de docker-compose

?

X

General

Débogage

General

Launch Action	Lancer le navigateur
Service Name	backend
Service URL	http://localhost
Target OS	Linux

Service URL

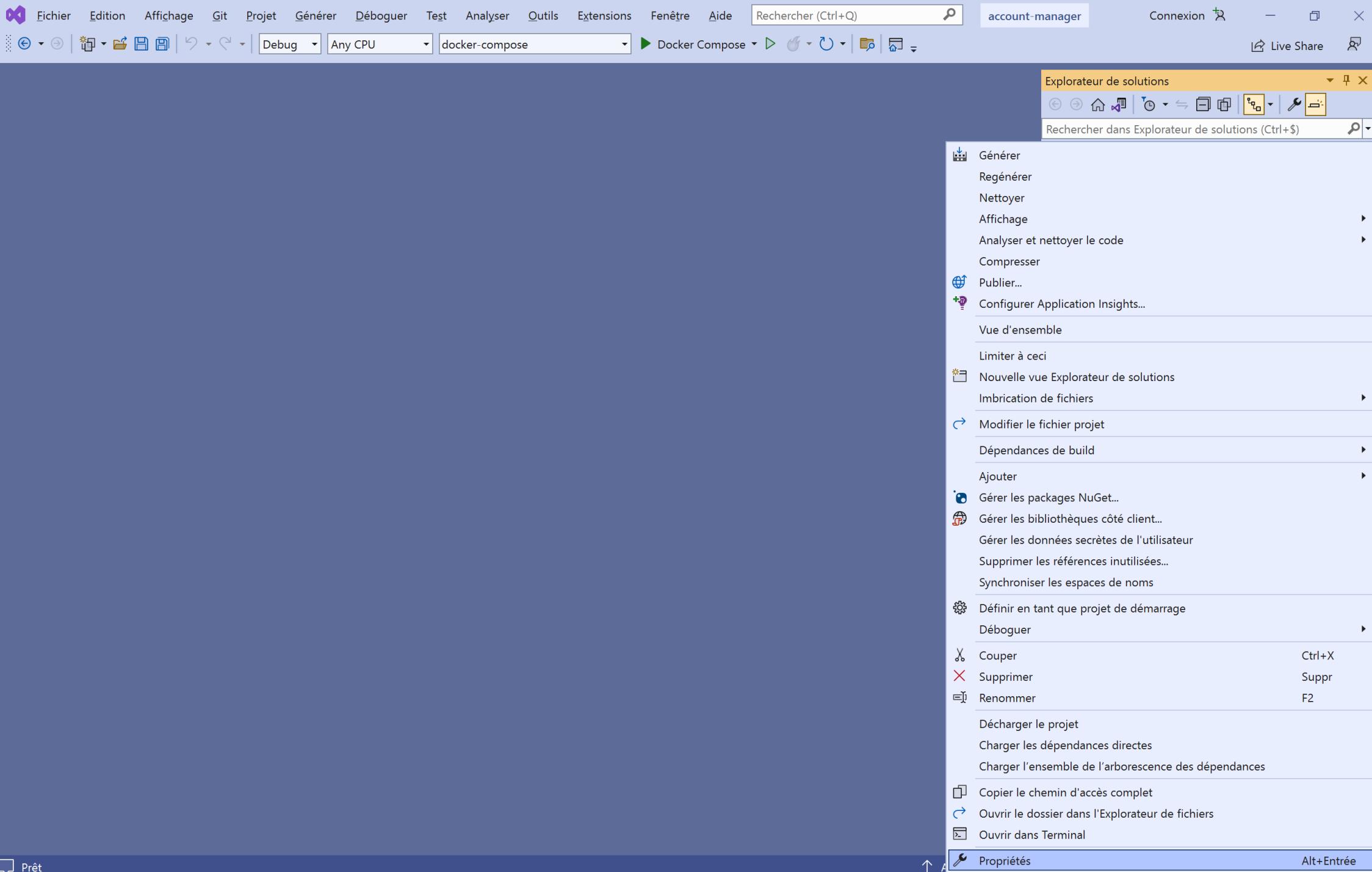
OK

Annuler

Appliquer

# 11

*Set backend launch settings*



Fichier Edition Affichage Git Projet Générer Déboguer Test Analyser Outils Extensions Fenêtre Aide Rechercher (Ctrl+Q) account-manager Connexion Live Share

backend

Rechercher parmi les propriétés

**Application**

Général

Ressources Win32

Création de packages

Utilisations globales

Build

Package

Code Analysis

Linting

Déboguer

Ressources

TypeScript

**Application**

**Type de sortie**  
Spécifie le type d'application à générer.  
**Application console**

**Framework cible** .NET 7.0

**SE cible** (Aucun)

**Objet de démarrage**  
Définit le point d'entrée à appeler au chargement de l'application. En règle générale, il s'agit du formulaire principal de votre application ou de la procédure 'Main' qui doit s'exécuter au démarrage de l'application. Les bibliothèques de classes ne définissent pas de point d'entrée.  
**(Non défini)**

**Nom d'assembly**  
Spécifie le nom du fichier de sortie qui doit contenir le manifeste de l'assembly.  
**\$(MSBuildProjectName)**

backend

**Espace de noms par défaut**  
Spécifie l'espace de noms de base des fichiers ajoutés au projet.  
**\$(MSBuildProjectName.Replace(" ", "\_"))**

Explorateur de solutions

Rechercher dans Explorateur de solutions (Ctrl+\$)

Solution 'account-manager' (3 sur 3 de projets)

- backend
  - Connected Services
  - Dépendances
  - Properties
  - Controllers
  - Services
  - appsettings.json
  - Dockerfile
  - C# Program.cs
  - C# WeatherForecast.cs
- docker-compose
  - .dockerignore
  - YML docker-compose.yml
    - docker-compose.override.yml
    - launchSettings.json
- frontend
  - npm
  - public
  - src
    - .gitignore
    - JS aspnetcore-https.js
    - JS aspnetcore-react.js
    - Dockerfile
    - nuget.config
    - package.json
    - package-lock.json
    - README.md

Fichier Edition Affichage Git Projet Générer Déboguer Test Analyser Outils Extensions Fenêtre Aide Rechercher (Ctrl+Q) account-manager Connexion Live Share

backend

Rechercher parmi les propriétés

Application Utilisations globales Build Package Code Analysis Linting Déboguer Général Profils de lancement de débogage Ressources Général Créer ou ouvrir des ressources d'assembly TypeScript Général Version ECMAScript Version ECMAScript à utiliser pour le code JavaScript généré ECMAScript 5 Mode de compilation pour les fichiers .tsx Spécifiez le mode de compilation du code JSX pour les fichiers .tsx. Ceci n'affecte pas les fichiers .ts Aucun Système de module Cible de génération de code de module externe Aucun

Déboguer

Général La gestion des profils de lancement a été déplacée vers une boîte de dialogue dédiée. Vous pouvez y accéder via le lien ci-dessous, via le menu Déboguer dans la barre de menus ou via la commande cible du débogage de la barre d'outils standard.

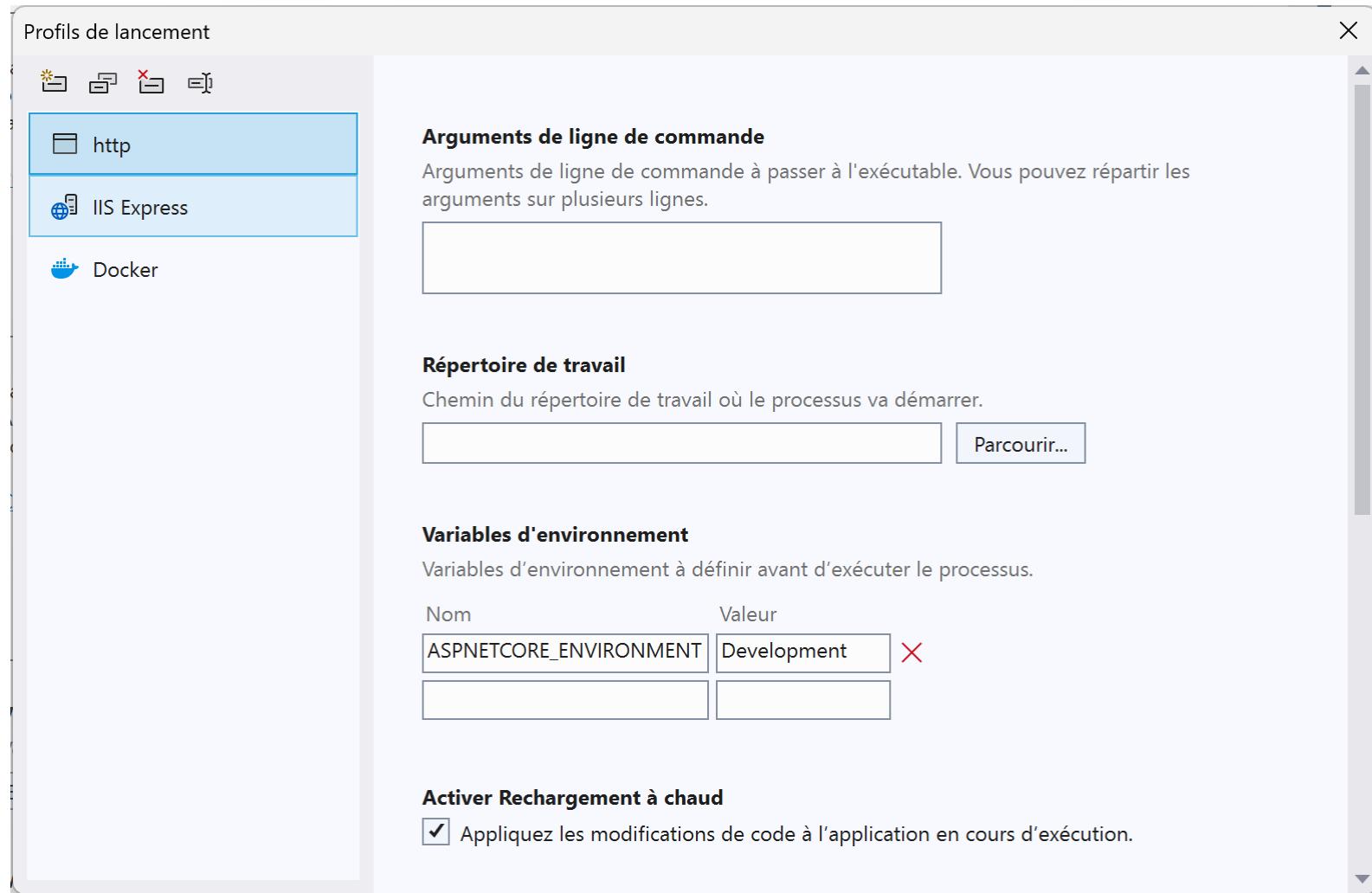
Explorateur de solutions

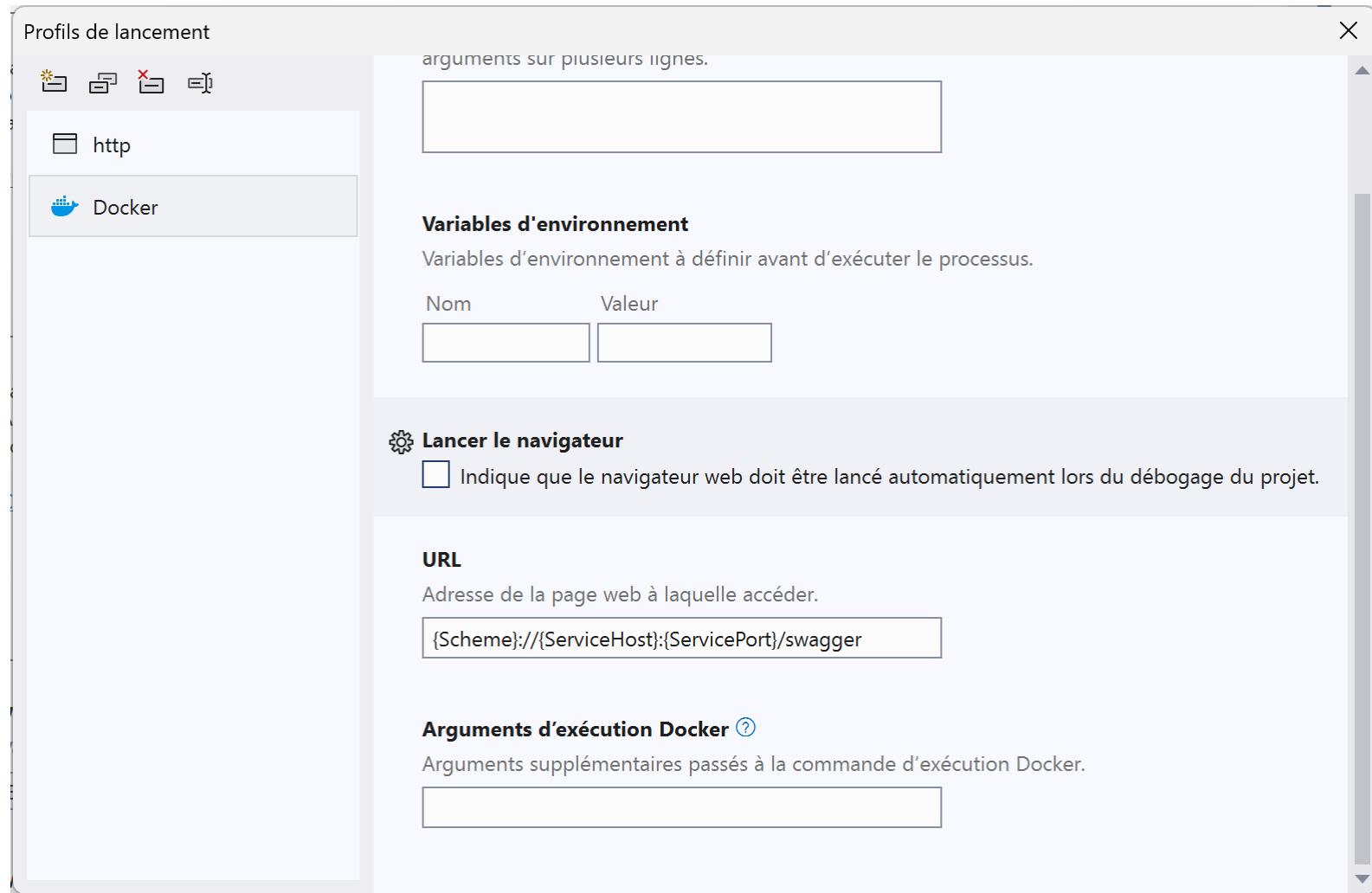
Solution 'account-manager' (3 sur 3 de projets)

- backend
  - Connected Services
  - Dépendances
  - Properties
  - Controllers
  - Services
  - appsettings.json
  - Dockerfile
  - C# Program.cs
  - C# WeatherForecast.cs
- docker-compose
  - .dockerignore
  - YML docker-compose.yml
    - docker-compose.override.yml
  - launchSettings.json
- frontend
  - npm
  - public
  - src
    - .gitignore
    - JS aspnetcore-https.js
    - JS aspnetcore-react.js
    - Dockerfile
    - nuget.config
    - package.json
    - package-lock.json
  - README.md

Rechercher dans Explorateur de solutions (Ctrl+\$)

Prêt Ajouter au contrôle de code source Sélectionner le référentiel





The screenshot shows the Visual Studio interface with the following details:

- Top Bar:** Fichier, Edition, Affichage, Git, Projet, Générer, Déboguer, Test, Analyser, Outils, Extensions, Fenêtre, Aide, Rechercher (Ctrl+Q), account-manager, Connexion.
- Toolbar:** Standard icons for file operations like Open, Save, Print, etc.
- Project Selector:** Debug, Any CPU, docker-compose.
- Status Bar:** 100 %, Aucun problème détecté, Lig. : 1, Car. : 1, SPC, CRLF.
- Solution Explorer:** Explorateur de solutions, showing the Solution 'account-manager' (3 sur 3 de projets) with three projects: backend, docker-compose, and frontend.
- Code Editor:** The main window displays the `launchSettings.json` file for the `backend` project. The schema is defined as `https://json.schemastore.org/launchsettings.json`. The code content includes profiles for HTTP, Docker, and IIS settings, specifying URLs and port numbers (e.g., `localhost:5185`, `localhost:27195`).
- Bottom Bar:** Explorateur de solutions, Modifications Git.

```
1 {
2   "profiles": {
3     "http": {
4       "commandName": "Project",
5       "launchBrowser": true,
6       "launchUrl": "swagger",
7       "environmentVariables": {
8         "ASPNETCORE_ENVIRONMENT": "Development"
9       },
10      "dotnetRunMessages": true,
11      "applicationUrl": "http://localhost:5185"
12    },
13    "Docker": {
14      "commandName": "Docker",
15      "launchUrl": "{Scheme}://{ServiceHost}:{ServicePort}/swagger",
16      "publishAllPorts": true
17    }
18  },
19  "$schema": "https://json.schemastore.org/launchsettings.json",
20  "iisSettings": {
21    "windowsAuthentication": false,
22    "anonymousAuthentication": true,
23    "iisExpress": {
24      "applicationUrl": "http://localhost:27195",
25      "sslPort": 0
26    }
27  }
28 }
```

The screenshot shows the Visual Studio interface with the following details:

- Top Bar:** Fichier, Edition, Affichage, Git, Projet, Générer, Déboguer, Test, Analyser, Outils, Extensions, Fenêtre, Aide, Rechercher (Ctrl+Q), account-manager, Connexion.
- Toolbar:** Standard icons for file operations like Open, Save, Print, etc.
- Project Selector:** Debug, Any CPU, docker-compose.
- Status Bar:** Lig. : 11, Car. : 47, SPC, CRLF.

**Solution Explorer:** Shows three projects: backend, docker-compose, and frontend.

- backend:** Contains Connected Services, Dépendances, Properties (with launchSettings.json selected), Controllers, Services, appsettings.json, Dockerfile, Program.cs, WeatherForecast.cs.
- docker-compose:** Contains .dockerignore, docker-compose.yml (selected), docker-compose.override.yml, launchSettings.json.
- frontend:** Contains npm, public, src, .gitignore, aspnetcore-https.js, aspnetcore-react.js, Dockerfile, nuget.config, package.json, package-lock.json, README.md.

**Code Editor:** The launchSettings.json file is open in the editor.

```
1 {  
2     "profiles": {  
3         "http": {  
4             "commandName": "Project",  
5             "launchBrowser": true,  
6             "launchUrl": "swagger",  
7             "environmentVariables": {  
8                 "ASPNETCORE_ENVIRONMENT": "Development"  
9             },  
10            "dotnetRunMessages": true,  
11            "applicationUrl": "http://localhost:5000"  
12        },  
13        "Docker": {  
14            "commandName": "Docker",  
15            "launchUrl": "{Scheme}://{ServiceHost}:{ServicePort}/swagger",  
16            "publishAllPorts": true  
17        }  
18    },  
19    "$schema": "https://json.schemastore.org/launchsettings.json"  
20}
```

**Bottom Status Bar:** 100 %, Aucun problème détecté, Explorateur de solutions, Modifications Git.

# 12

Set *frontend* proxy settings  
(for standalone development mode)

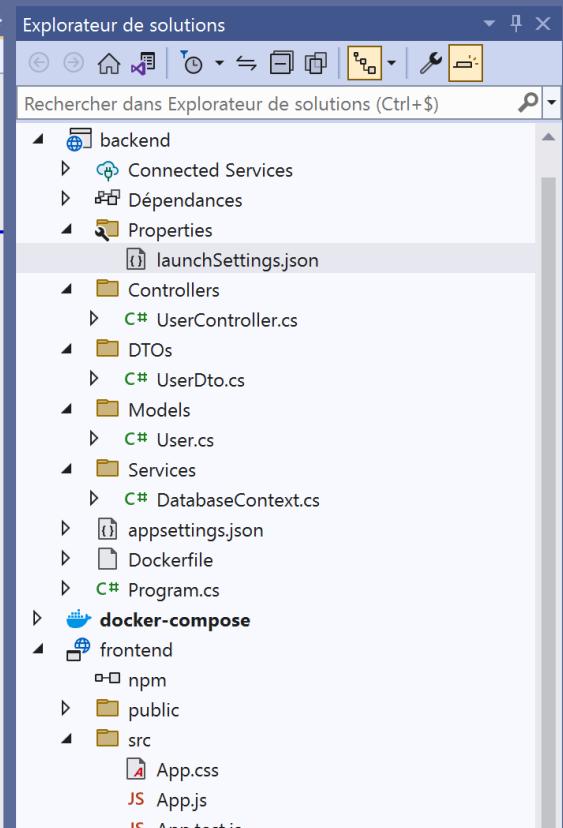
The screenshot shows a Visual Studio interface with the following details:

- Top Bar:** Fichier, Edition, Affichage, Git, Projet, Générer, Déboguer, Test, Analyser, Outils, Extensions, Fenêtre, Aide, Rechercher (Ctrl+Q), account-manager, Connexion.
- Toolbar:** Standard icons for file operations like Open, Save, Print, etc.
- Project Explorer:** Shows a Docker Compose project structure:
  - backend
  - Connected Services
  - Dépendances
  - Properties
    - launchSettings.json
  - Controllers
  - Services
  - appsettings.json
  - Dockerfile
  - C# Program.cs
  - C# WeatherForecast.cs
  - docker-compose
  - .dockerignore
  - YML docker-compose.yml
    - docker-compose.override.yml
    - launchSettings.json
  - frontend
  - npm
  - public
  - src
    - App.css
    - JS App.js
    - JS App.test.js
    - index.css
    - JS index.js
    - logo.svg
    - JS reportWebVitals.js
    - JS setupProxy.js
    - JS setupTests.js
    - .gitignore
    - JS aspnetcore-https.js
    - JS aspnetcore-react.js
    - Dockerfile
    - nuget.config
    - package.json
    - package-lock.json
    - README.md

- Solution Explorer:** Displays the same project structure as the Project Explorer.
- Code Editor:** The setupProxy.js file is open in the editor, showing code to set up a proxy middleware for a weather forecast application.
- Status Bar:** 100 %, Aucun problème détecté, Lig. : 1, Car. : 1, SPC, CRLF.
- Bottom Bar:** Explorateur de solutions, Modifications Git.

Divers Program.cs

```
1 const { createProxyMiddleware } = require('http-proxy-middleware');
2
3 const context = [
4   '/api',
5 ];
6
7 module.exports = function (app) {
8   const appProxy = createProxyMiddleware(context, {
9     target: 'http://localhost:5000',
10    secure: false
11  });
12
13  app.use(appProxy);
14};
15
```



100 % Aucun problème détecté Lig. : 9 Car. : 39 SPC CRLF

Sortie

Afficher la sortie à partir de : Outils de conteneur

```
Container database Created
Container backend Creating
Container backend Created
Container frontend Starting
Container reverse-proxy Starting
Container database Starting
Container reverse-proxy Started
Container frontend Started
Container database Started
Container backend Starting
Container backend Started
Terminé ! Les conteneurs Docker sont prêts.
```

# 13

Link *frontend* to *backend*

Fichier Edition Affichage Git Projet Générer Déboguer Test Analyser Outils Extensions Fenêtre Aide Rechercher (Ctrl+Q) account-manager Connexion Live Share

App.js ▾ X setupProxy.js

Divers renderForecastsTable forecasts.map() callback

```
1 import React, { Component } from 'react';
2
3 export default class App extends Component {
4     static displayName = App.name;
5
6     constructor(props) {
7         super(props);
8         this.state = { forecasts: [], loading: true };
9     }
10
11    componentDidMount() {
12        this.populateWeatherData();
13    }
14
15    static renderForecastsTable(forecasts) {
16        return (
17            <table className='table table-striped' aria-labelledby="tabelLabel">
18                <thead>
19                    <tr>
20                        <th>Date</th>
21                        <th>Temp. (C)</th>
22                        <th>Temp. (F)</th>
23                        <th>Summary</th>
24                    </tr>
25                </thead>
26                <tbody>
27                    {forecasts.map(forecast =>
28                        <tr key={forecast.date}>
29                            <td>{forecast.date}</td>
30                            <td>{forecast.temperatureC}</td>
31                            <td>{forecast.temperatureF}</td>
32                            <td>{forecast.summary}</td>
33                        </tr>
34                    )}
35                </tbody>
36            </table>
37        );
38    }
39
40    render() {
41        let contents = this.state.loading
42            ? <p><em>Loading... Please refresh once the ASP.NET backend has started. See <a href="https://aka.ms/jspintegrationreact">https://aka.ms/jspintegrationreact</a></em></p>
43            : App.renderForecastsTable(this.state.forecasts);
44
45        return (
46            <div>
47                <h1 id="tabelLabel" >Weather forecast</h1>
48                <p>This component demonstrates fetching data from the server.</p>
49            </div>
50        );
51    }
52}
```

Explorateur de solutions

Rechercher dans Explorateur de solutions (Ctrl+\$)

- Solution 'account-manager' (3 sur 3 de projets)
  - backend
  - docker-compose
  - frontend
    - npm
    - public
    - src
      - App.css
      - JS App.js
      - JS App.test.js
      - index.css
      - JS index.js
      - logo.svg
      - JS reportWebVitals.js
      - JS setupProxy.js
      - JS setupTests.js
    - .gitignore
    - JS aspnetcore-https.js
    - JS aspnetcore-react.js
    - Dockerfile
    - nuget.config
    - package.json
    - package-lock.json
    - README.md

91 % Aucun problème détecté

Lig. : 16 Car. : 17 SPC CRLF

Explorateur de solutions Modifications Git

Prêt Ajouter au contrôle de code source Sélectionner le référentiel

Fichier Edition Affichage Git Projet Générer Déboguer Test Analyser Outils Extensions Fenêtre Aide Rechercher (Ctrl+Q) account-manager Connexion Live Share

App.js\* setupProxy.js

Divers renderForecastsTable forecasts.map() callback

```
1 import React, { Component } from 'react';
2
3 export default class App extends Component {
4   static displayName = App.name;
5
6   constructor(props) {
7     super(props);
8     this.state = { users: [], loading: true };
9   }
10
11   componentDidMount() {
12     this.populateWeatherData();
13   }
14
15   static renderForecastsTable(forecasts) {
16     return (
17       <table className='renderForecastsTable' ^>
18         <thead>
19           <tr>
20             <th>Date</th>
21             <th>Temperature (C)</th>
22             <th>Temperature (F)</th>
23             <th>Summary</th>
24           </tr>
25         </thead>
26         <tbody>
27           {forecasts.map(forecast =>
28             <tr key={forecast.date}>
29               <td>{forecast.date}</td>
30               <td>{forecast.temperatureC}</td>
31               <td>{forecast.temperatureF}</td>
32               <td>{forecast.summary}</td>
33             </tr>
34           )}
35         </tbody>
36       </table>
37     );
38   }
39
40   render() {
41     let contents = this.state.loading
42       ? <p><em>Loading... Please refresh once the ASP.NET backend has started. See <a href="https://aka.ms/jspintegrationreact">https://aka.ms/jspintegrationreact</a></em></p>
43       : App.renderForecastsTable(this.state.forecasts);
44
45     return (
46       <div>
47         <h1 id="tabelLabel" >Weather forecast</h1>
48         <p>This component demonstrates fetching data from the server.</p>
49       </div>
50     );
51   }
52 }
```

Le renommage va entraîner la mise à jour de 2 références dans 1 fichier.

Inclure les commentaires Inclure les chaînes Entrer pour renommer, Maj+Entrée pour afficher un aperçu.

Explorateur de solutions

Rechercher dans Explorateur de solutions (Ctrl+\$)

Solution 'account-manager' (3 sur 3 de projets)

- backend
- docker-compose
- frontend
  - npm
  - public
  - src
    - App.css
    - App.js
    - App.test.js
    - index.css
    - index.js
    - logo.svg
    - reportWebVitals.js
    - setupProxy.js
    - setupTests.js
  - .gitignore
  - aspnetcore-https.js
  - aspnetcore-react.js
  - Dockerfile
  - nuget.config
  - package.json
  - package-lock.json
  - README.md

91 % Aucun problème détecté

Lig. : 15 Car. : 32 SPC CRLF

↑ Ajouter au contrôle de code source ▾ Sélectionner le référentiel ▾

```
static renderUsers(users) {
  return (
    <table className='table table-striped' aria-labelledby="tableLabel">
      <thead>
        <tr>
          <th>Name</th>
          <th>Email</th>
        </tr>
      </thead>
      <tbody>
        {users.map(user =>
          <tr key={user.name}>
            <td>{user.name}</td>
            <td>{user.email}</td>
          </tr>
        )}
      </tbody>
    </table>
  );
}
```

Eichier Edition Affichage Git Projet Générer Déboguer Test Analyser Outils Extensions Fenêtre Aide Rechercher (Ctrl+Q) account-manager Connexion Live Share

App.js Divers users.map() callback

```
1 import React, { Component } from 'react';
2
3 export default class App extends Component {
4     static displayName = App.name;
5
6     constructor(props) {
7         super(props);
8         this.state = { users: [], loading: true };
9     }
10
11     componentDidMount() {
12         this.populateUserData();
13     }
14
15     static renderUsers(users) {
16         return (
17             <table className='table table-striped' aria-labelledby="tabelLabel">
18                 <thead>
19                     <tr>
20                         <th>Name</th>
21                         <th>Email</th>
22                     </tr>
23                 </thead>
24                 <tbody>
25                     {users.map(user =>
26                         <tr key={user.name}>
27                             <td>{user.name}</td>
28                             <td>{user.email}</td>
29                         </tr>
30                     )}
31                 </tbody>
32             </table>
33         );
34     }
35
36     render() {
37         let contents = this.state.loading
38             ? <p><em>Loading... Please refresh once the ASP.NET backend has started. See <a href="https://aka.ms/jspintegrationreact">https://aka.ms/jspintegrationreact</a></em></p>
39             : App.renderUsers(this.state.users);
40
41         return (
42             <div>
43                 <h1 id="tabelLabel" >Users</h1>
44                 <p>This component demonstrates fetching data from the server.</p>
45                 {contents}
46             </div>
47         );
48     }
49
50     async populateUserData() {
51         const response = await fetch('api/users');
52         const data = await response.json();
53         // console.log(data);
54         this.setState({ users: data, loading: false });
55     }
56 }
```

75% Aucun problème détecté

Docker Compose

Explorateur de solutions

Rechercher dans Explorateur de solutions (Ctrl+\$)

Solution 'account-manager' (3 sur 3 de projets)

- backend
- docker-compose
- frontend
  - npm
  - public
  - src
    - App.css
    - App.js
    - App.test.js
    - index.css
    - index.js
    - logo.svg
    - reportWebVitals.js
    - setupProxy.js
    - setupTests.js
  - .gitignore
  - aspnetcore-https.js
  - aspnetcore-react.js
  - Dockerfile
  - nuget.config
  - package.json
  - package-lock.json
  - README.md

Lig. : 28 Car. : 40 SPC CRLF

Explorateur de solutions Modifications Git

Éléments enregistrés Ajouter au contrôle de code source Sélectionner le référentiel

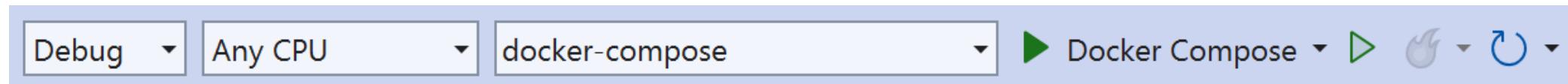
14

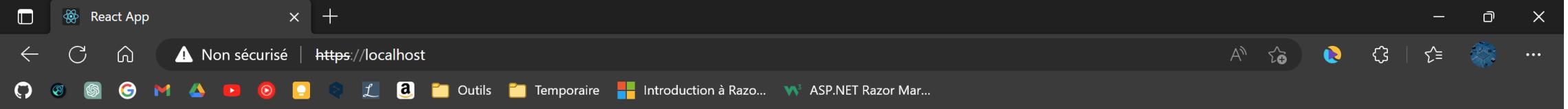
Start *docker-compose* project

## Production mode :

```
PS C:\Users\Kevin> cd .\source\repos\account-manager\  
PS C:\Users\Kevin\source\repos\account-manager> docker-compose up -d --build
```

Debug mode :





# Users

This component demonstrates fetching data from the server.

Name	Email
string	string

React App    Swagger UI

Non sécurisé | https://localhost/swagger/index.html

Outils Temporaire Introduction à Razor... ASP.NET Razor Mar...

Swagger. Supported by SMARTBEAR

Select a definition backend v1

# backend

1.0 OAS3

<https://localhost/swagger/v1/swagger.json>

## User

GET /api/users

POST /api/users

GET /api/users/{id}

PUT /api/users/{id}

DELETE /api/users/{id}

## Schemas

UserGetDto >

UserPostDto >

