

Mini-projet de synthèse

Dans une entreprise de services, les utilisateurs signalent des pannes techniques (connexion lente, bugs d'applications, etc.). Vous devez créer un microservice **SupportService** pour enregistrer et consulter les réclamations techniques.

1. Créez une solution Visual Studio `SupportSolution` et ajoutez un projet Web API nommé `SupportService`
2. Créez une classe `Reclamation.cs` dans un dossier `Models`, Cette entité doit contenir les informations suivantes (3 pts)
 - Un identifiant unique (`Id`) de type entier.
 - Un sujet (`Sujet`) de la réclamation, sous forme de texte.
 - Une description (`Description`) détaillée de la réclamation, sous forme de texte.
3. Dans une application ASP.NET Core utilisant Entity Framework Core, vous devez configurer l'accès à la base de données à l'aide d'un contexte de données.
Créez une classe `ApplicationDbContext.cs` dans un dossier nommé `Data`
 - Cette classe doit hériter de `DbContext` et disposer d'un constructeur prenant en paramètre un objet `DbContextOptions<ApplicationDbContext>`. Grâce à cela, toutes les méthodes du contrôleur peuvent accéder à la base de données via `_context`.
 - Ajoutez une propriété de type `DbSet<Reclamation>` pour gérer les opérations CRUD sur les réclamations.
4. Créez un contrôleur `ReclamationsController.cs` avec 2 actions
 - GET `/reclamations` → retourne toutes les réclamations
 - POST `/reclamations` → ajoute une nouvelle réclamation
5. L'injection de dépendances : au lieu de créer un objet `ApplicationDbContext` à la main dans chaque contrôleur, on déclare dans `Program.cs` que ce service doit être géré automatiquement par ASP.NET Core.
Ajoutez l'injection de dépendance du contexte de base de données `ApplicationDbContext` dans le fichier `Program.cs`. (l'ajout doit être fait avant : `var app = builder.Build();`)
6. Effectuez la migration et création de la base (Installer l'outil `dotnet ef` (si pas encore installé))
7. Testez ton contrôleur `ReclamationsController` avec le fichier `SupportSolution.http` qui permet de tester facilement les endpoints de l'API sans navigateur :

```
@SupportSolution_HostAddress = http://localhost:5256
### Cette requête appelle l'endpoint /reclamations avec la méthode GET pour
récupérer toutes les réclamations.
GET {{SupportSolution_HostAddress}}/reclamations
Accept: application/json
###
### Cette requête envoie une nouvelle réclamation à l'API via la méthode POST.
POST {{SupportSolution_HostAddress}}/reclamations
Content-Type: application/json
{
  "sujet": "Connexion lente",
  "description": "L'application met du temps à charger."
}
###
```