Guide Complet pour la Création d'une API avec ASP.NET Core et SQL Server

Introduction

Ce guide détaille les étapes nécessaires à la création d'une API RESTful avec ASP.NET Core et à l'utilisation de SQL Server pour la gestion de la base de données. Vous suivrez un processus étape par étape depuis la création du projet jusqu'au déploiement en production.

Prérequis

Avant de commencer, assurez-vous d'avoir installé les outils suivants :

- Visual Studio (version recommandée)
- SQL Server Management Studio (SSMS)
- .NET Core SDK
- Postman (ou un outil similaire pour tester les API)

Étape 1 : Création du Projet ASP.NET Core

1.1 Installation et Configuration de l'Environnement de Développement

- Téléchargez et installez Visual Studio depuis le site officiel de Microsoft.
- Installez le .NET Core SDK depuis le site officiel de .NET.

1.2 Création du Projet ASP.NET Core Web API

- 1. Lancez Visual Studio.
- 2. Sélectionnez "Créer un nouveau projet".
- 3. Choisissez "Projet ASP.NET Core Web API" et nommez-le selon votre préférence.
- 4. Configurez le projet avec les options par défaut.

Étape 2 : Configuration de la Base de Données SQL Server

2.1 Installation de SQL Server et SSMS

- Téléchargez et installez SQL Server depuis <u>le site officiel de Microsoft</u>.
- Installez SQL Server Management Studio (SSMS) pour une gestion facilitée de la base de données.

2.2 Création de la Base de Données

- 1. Lancez SQL Server Management Studio (SSMS).
- 2. Connectez-vous à votre serveur SQL local.
- 3. Créez une nouvelle base de données nommée [NomDeVotreBaseDeDonnées].

2.3 Création des Tables et des Migrations

- 1. Utilisez les migrations pour créer vos tables à partir de vos modèles.
 - o Exemple de modèle : User.cs

```
csharp
Copier le code
public class User
{
    public int Id { get; set; }
    public string Name { get; set; }
    public string Email { get; set; }
    // Ajoutez d'autres propriétés au besoin
}
```

2. Exécutez les commandes CLI suivantes dans votre terminal pour créer et appliquer des migrations :

```
bash
Copier le code
dotnet ef migrations add InitialCreate
dotnet ef database update
```

Étape 3 : Développement de l'API avec ASP.NET Core

3.1 Création du Contexte de Base de Données

• Créez un fichier AppDbContext.cs pour définir le contexte de base de données :

```
csharp
Copier le code
public class AppDbContext : DbContext
{
    public AppDbContext(DbContextOptions<AppDbContext> options) :
base(options) { }

    public DbSet<User> Users { get; set; }
    // Ajoutez d'autres DbSets pour d'autres modèles
}
```

3.2 Création des Contrôleurs API

• Créez un contrôleur UsersController.cs pour gérer les opérations CRUD sur les utilisateurs :

```
csharp
```

```
Copier le code
[Route("api/[controller]")]
[ApiController]
public class UsersController : ControllerBase
{
    private readonly AppDbContext _context;

    public UsersController(AppDbContext context)
    {
        _context = context;
}

    // Implémentez les méthodes GET, POST, PUT, DELETE pour gérer les utilisateurs
}
```

3.3 Configuration des Services et Injection de Dépendances

• Configurez les services dans Startup.cs:

```
csharp
Copier le code
public void ConfigureServices(IServiceCollection services)
{
    services.AddDbContext<AppDbContext>(options =>

    options.UseSqlServer(Configuration.GetConnectionString("DefaultConnection")));
    services.AddControllers();
}
```

Étape 4 : Test et Déploiement

4.1 Test de l'API avec Postman

• Utilisez Postman pour tester les endpoints de votre API :

```
o GET /api/users
o POST /api/users
o PUT /api/users/{id}
o DELETE /api/users/{id}
```

4.2 Déploiement de l'API

 Déployez votre application ASP.NET Core sur une plateforme comme Azure, Heroku ou AWS.

Conclusion

Ce guide vous a accompagné à travers les étapes essentielles pour créer une API robuste avec ASP.NET Core et SQL Server. Assurez-vous de documenter vos modifications, de tester rigoureusement votre API et de sécuriser votre base de données pour garantir la qualité et la sécurité de votre application.

Ce document est conçu comme un guide général. Assurez-vous d'adapter les exemples et les instructions en fonction des besoins spécifiques de votre projet et de votre environnement de développement.

Vous pouvez copier-coller ce contenu dans un fichier Word et le sauvegarder en tant que fichier .docx selon vos besoins. Assurez-vous de personnaliser les sections telles que [NomDeVotreBaseDeDonnées], les exemples de modèles, et d'ajuster les détails spécifiques à votre projet.

3.5