# Mode d’emploi – Déploiement et utilisation de l’application ArtGallery

Ce guide explique comment déployer et utiliser l’application ArtGallery avec ASP.NET Core, SQL Server (SSMS) et Entity Framework Core.

## Architecture de l’application (NOUVEAUTÉ)

L’application ArtGallery applique désormais une architecture propre (Clean Architecture) :

* **Séparation stricte des responsabilités** : la logique métier (CRUD, règles, gestion d’images, etc.) est déplacée dans des classes de service dédiées.
* **Services métiers** : chaque entité principale dispose d’un service :
  + ArtworkService (œuvres)
  + CategoryService (catégories)
  + ExhibitionService (expositions)
* **Interfaces** : chaque service implémente une interface (IArtworkService, ICategoryService, IExhibitionService) pour garantir la cohérence et faciliter les tests.
* **Injection de dépendances (DI)** : les services sont enregistrés dans le conteneur DI dans Program.cs :
* builder.Services.AddScoped<IArtworkService, ArtworkService>();  
  builder.Services.AddScoped<ICategoryService, CategoryService>();  
  builder.Services.AddScoped<IExhibitionService, ExhibitionService>();
* **Contrôleurs simplifiés** : les contrôleurs ne contiennent plus de logique métier. Ils orchestrent simplement les appels aux services et passent les données aux vues.

**Avantages :** - Code plus maintenable, évolutif et testable - Respect des bonnes pratiques ASP.NET Core - Ajout de nouvelles règles métier ou entités facilité

## 1. Prérequis

* **Windows 10/11**
* **.NET 6 SDK** ou version compatible
* **SQL Server** (Express, Developer ou autre) + **SQL Server Management Studio (SSMS)**
* **Visual Studio 2022** (ou VS Code avec extensions C#)
* **Git** (optionnel, pour cloner le dépôt)

## 2. Préparation de la base de données

1. **Démarrer SQL Server** (via SSMS ou en tant que service Windows).
2. **Créer une base de données** (ex: ArtGalleryDB) :
   * Ouvrir SSMS
   * Se connecter à l’instance locale
   * Clic droit sur “Bases de données” > “Nouvelle base de données…”
   * Nommer la base (ex: ArtGalleryDB) et valider
3. **Récupérer la chaîne de connexion** :
   * Format général :
   * Server=localhost;Database=ArtGalleryDB;Trusted\_Connection=True;
   * Adapter selon votre configuration (authentification Windows ou SQL Server)

## 3. Configuration de l’application

1. **Ouvrir le projet dans Visual Studio**
2. **Configurer la chaîne de connexion** dans appsettings.json :

* "ConnectionStrings": {  
   "DefaultConnection": "Server=localhost;Database=ArtGalleryDB;Trusted\_Connection=True;"  
  }

1. **Vérifier les paramètres de migration** (le projet doit référencer Entity Framework Core et avoir les fichiers de migration si existants)

## 4. Création et migration de la base de données

1. **Ouvrir la Console du Gestionnaire de Package NuGet** (Outils > Gestionnaire de package NuGet > Console du gestionnaire de package)
2. **Exécuter les commandes suivantes** :
   * Pour créer la base (si migrations présentes) :
   * Update-Database
   * Pour générer une migration (si nécessaire) :
   * Add-Migration InitialCreate  
     Update-Database

## 5. Lancement de l’application

1. **Lancer le projet** (F5 ou bouton “Démarrer”)
2. L’application s’ouvre dans le navigateur à l’adresse https://localhost:xxxx (le port peut varier)
3. **Créer un compte administrateur** à la première connexion (ou via SSMS si besoin)

## 6. Utilisation de l’application

* **Navigation** :
  + Accueil, Galerie, Expositions, Catégories, Panier, Commandes, Livre d’or, Liens externes
* **Gestion** (admin) :
  + Ajout/modification/suppression d’œuvres, expositions, catégories, liens, validation du livre d’or
* **Achat** (client) :
  + Ajouter au panier, commander, suivre l’historique
* **Sécurité** :
  + Authentification requise pour commander ou gérer le contenu

## 7. Dépannage

* **Erreur de connexion à la base** :
  + Vérifier la chaîne de connexion, que SQL Server est bien démarré, que l’utilisateur a les droits
* **Problème de migration** :
  + Vérifier les dépendances NuGet (Microsoft.EntityFrameworkCore.\*)
  + Nettoyer/reconstruire la solution
* **Erreur d’injection de dépendance (DI) pour un service** :
  + Vérifier que les services sont bien enregistrés dans Program.cs (voir section “Architecture de l’application”)
  + Vérifier que les fichiers d’interface et de service sont bien présents dans le dossier Services
  + Vérifier les using en haut des contrôleurs (using ArtGallery.Services;)
* **Autres** :
  + Consulter la console de sortie de Visual Studio pour plus de détails

## 8. Ressources utiles

* Documentation officielle ASP.NET Core : https://learn.microsoft.com/fr-fr/aspnet/core/
* Documentation Entity Framework Core : https://learn.microsoft.com/fr-fr/ef/core/
* Documentation SQL Server : https://learn.microsoft.com/fr-fr/sql/

*Pour toute question ou problème, contacter l’administrateur du projet ou consulter la documentation technique jointe.*