

Module .NET

UFR Sciences et Techniques

Université de Rouen

Master 2 GIL

FTEL

TP2

Durée 4h



Installation de SQL Server Express

1. Téléchargez et installez SQL Server Express - <https://www.microsoft.com/fr-fr/sql-server/sql-server-editions-express>
2. (Optionnel) Téléchargez et installez SQL Server Management Studio - <https://docs.microsoft.com/fr-fr/sql/ssms/download-sql-server-management-studio-ssms>

En attendant...

1. Vous pouvez continuer le TP précédant pour ajouter des fonctionnalités à votre projet

Ajouter EntityFramework

1. Ajoutez-le package NuGet « EntityFramework »
 - a. Clic droit sur votre projet
 - b. Gérer les packages NuGet
 - c. Onglet Parcourir
2. Une fois cela fait, vérifiez que le web.config a bien été modifié
 - a. La configSection entityFramework doit être présente
 - b. La section entityFramework doit être présente
3. Créez votre connectionString dans le web.config

```
<connectionStrings>
  <add name="M2LinkContext" connectionString="Server=.\SQLEXPRESS;Initial
Catalog=M2Link_Dev; Integrated Security=True;
multipleactiveresultsets=True;App=EntityFramework"
providerName="System.Data.SqlClient"/>
</connectionStrings>
```

- a. Si vous utilisez un SQL non Express, vous devez modifier le connectionString pour mettre « Server=. »

Le Context

1. Créez dans un dossier Context votre classe M2LinkContext
 - a. Elle doit dériver de DbContext
 - b. Elle doit appeler le constructeur de DbContext avec le nom de votre ConnectionString en paramètre « M2LinkContext »

Vos entités

1. Créez un dossier Entities
2. Ajouter une classe « User »
3. Ajoutez les attributs de votre classe User
 - a. N'oubliez pas de mettre un Id de type « Guid » et annoté de « KeyAttribute »
4. Liez votre « User » à votre « Context »
 - a. En utilisant un DbSet

Vos repositories

1. Créez un dossier « Repositories »
2. Créez une classe « UserRepository »
 - a. Qui contient un constructeur qui prend en paramètre votre context
3. Créez une méthode « Add »
 - a. Qui prend en paramètre un « User »
4. Créez une méthode « GetAll »
 - a. Qui retourne tous les « User »
5. Supprimez votre ancienne classe « Database » et utilisez votre « Context » et votre « UserRepository » pour requêter vos « User » en base de données
 - a. Pensez à utiliser « using »
 - b. N'oubliez pas de sauvegarder vos changements

Première requête

1. Générez une première utilisation de votre « Repository » afin que Entity génère votre base de données

Explorateur d'objets SQL Server

1. Ouvrez la fenêtre « Explorateur d'objets SQL Server » de Visual Studio
2. Ajouter une connexion
 - a. Nom du serveur : « .\SQLExpress »
3. Vérifiez que votre base de données a bien été créée
4. Vérifiez que votre table a bien été créée
5. Vérifiez que vos données sont bien présentes

SQL Server Management Studio (Optionnel)

1. Ouvrez SQL Server Management Studio
2. Connectez-vous à votre instance SQLExpress
 - a. .\SQLEXPRESS
3. Vérifiez que votre base de données a bien été créée
4. Vérifiez que votre table a bien été créée
5. Vérifiez que vos données sont bien présentes

DatabaseInitializer

1. Dans votre dossier « Context », créez une classe « M2LinkInitializer »
 - a. Elle doit dériver de « DropCreateDatabaseIfModelChanges<M2LinkContext> »
2. Surchargez la méthode « Seed »
3. Initialisez des données par défaut
4. Ajouter votre initialiseur à votre web.config

```
<entityFramework>
  <contexts>
    <context type="M2Link.Context.M2LinkContext, M2Link">
      <databaseInitializer type="M2Link.Context.M2LinkInitializer, M2Link" />
    </context>
  </contexts>
</entityFramework>
```

5. Supprimer votre base de données

6. Rechargez votre application
7. Vérifiez que vos données sont bien en base de données

Gestion de l'authentification

1. Créez une page de connexion
 - a. Avec vérification du pseudo / mot de passe
 - b. Si valide, redirigez vers l'accueil
 - c. Sinon, réaffichez la page avec les messages d'erreur
2. Ajouter l'authentification avant de rediriger vers l'accueil
 - a. Pensez à utiliser « FormsAuthentication.SetAuthCookie »
3. Empêchez l'accès à la page d'accueil sans être authentifié
 - a. Pensez à utiliser « [Authorize] »
 - b. Pensez à renseigner le mode d'authentification et l'url de la page de login dans le web.config

```
<system.web>  
  <authentication mode="Forms">  
    <forms loginUrl="~/Login/Form" />  
  </authentication>  
</system.web>
```

4. Ajouter un bouton de déconnexion dans le menu

Edition de son profil

1. Créez une page permettant d'éditer son profil
 - a. Créez un modèle
 - b. Affichez les anciennes valeurs
 - c. Sauvegarder les données

Affichage des utilisateurs

1. Créer une page pour afficher l'ensemble des utilisateurs inscrits sur votre M2Link
 - a. Il faut être authentifié pour y accéder
2. Créer une page de profil pour afficher un profil d'utilisateur
 - a. Il faut être authentifié pour y accéder

Poster un message

1. Créez votre entité « Message »
 - a. Un « User » a plusieurs « Message »
 - b. Un « Message » a un « User »
 - c. N'oubliez pas le virtual
2. Faites le formulaire pour qu'un utilisateur connecté puisse poster un message
3. Faites la page pour qu'un utilisateur puisse voir ses messages dans son profil

Suivre un utilisateur

1. Faites la fonctionnalité pour qu'un utilisateur puisse suivre d'autres utilisateurs

Affichage des messages des utilisateurs

1. Faites la fonctionnalité pour qu'un utilisateur puisse voir, sur la page d'accueil connecté, l'ensemble des messages des utilisateurs qu'il suit

Pour aller plus loin

1. Ajoutez une photo aux profils des utilisateurs
2. Ajoutez une description aux profils des utilisateurs
3. Chiffrez les mots de passe
4. Mettez en forme les pages
 - a. Pensez à utiliser le Layout
5. Améliorez les messages pour mettre des ♥
6. Améliorez les messages pour mettre une image