

Modalités

Vous devez rendre ce projet en m'envoyant par mail (cguquet@synexie.fr) le lien du repository Git de votre choix (GitHub, GitLab, ...).

Vous prendrez soin d'utiliser un fichier .gitignore adéquat afin de ne pas inclure dans les sources les fichiers issus de la compilation.

Conseils

Vous pouvez (et c'est même recommandé) partir de la structure de projet qui a été réalisée en cours, et toujours disponible sur GitHub à cette adresse : <https://github.com/kall2sollies/lSen2019-11> en prenant soin de conserver tout ce qui est générique et retirer ce qui est spécifique au modèle de données établi en cours.

Vous pouvez bien évidemment chercher sur internet toutes les ressources qui vous paraissent nécessaires (aide sur Stack Overflow, librairies sur Github, Nuget, npm...).

En d'autres termes, la réalisation du projet est évaluée sur le plan de sa réussite et de son efficacité, donc plutôt de façon professionnelle que scolaire.

Notation

Un code qui compile et s'exécute directement sans devoir modifier de paramètres d'environnement (chemins de fichiers par exemple) sera très apprécié !

Cependant, un projet rendu non fonctionnel (non compilable, ou plantant en runtime) sera quand même noté, le code rédigé étant lu et apprécié, quel que soit son aboutissement, par rapport à ce qui est demandé dans l'énoncé (ceci, par contre, est bien plus scolaire que professionnel).

Sujet

L'objectif de ce projet est de réaliser un ensemble de formulaires permettant de gérer une liste de personnes, de leur affecter un service (ex : marketing, production, ...), et de leur affecter des rôles (exemple : utilisateur, manager, administrateur, superadministrateur...).

Ce projet sera très similaire à ce qui a été fait en cours, à la différence près que entre un utilisateur et un rôle, il y a une relation n..n (un utilisateur peut avoir plusieurs rôles, et chaque rôle peut être affecté à plusieurs utilisateurs).

Vous devrez donc chercher sur internet comment modéliser les relations n..n (appelées aussi many-to-many) avec Entity Framework Core.

Question 1 : Modèle (5 pts)

Créer le modèle objet correspondant aux descriptions suivantes (la description est fonctionnelle, libre à vous de modéliser toutes les classes qui vous paraissent utiles).

- Une personne
 - A un id, un nom, un prénom, une date de naissance, un téléphone, une adresse mail
 - Appartient à un service (relation 0..1)
 - A une liste de rôles (relation n..n)
- Un service a un nom et un id
- Un rôle a un id, un nom, et une liste de personnes (relation n..n)

Question 2 : Formulaire d'édition simples (5 pts)

Sur le modèle de ce qui a été réalisé en cours, créer les contrôleurs, vues liste, vues détail et entrées de menu permettant de gérer les modèles suivants :

- Personne (sans les relations)
- Service
- Rôle (sans les relations)

Vous pouvez, si vous le souhaitez, afin de mutualiser du code, implémenter le pattern du repository (ce qui donne 2 pts de bonus)

Question 3 : Imports (5 pts)

En réutilisant le principe de d'initialisation des données de test vu en cours, vous devrez importer des données de test pour chaque modèle (personnes, services, rôles, contacts, types)

Question 4 : Formulaire avec relations (5 pts)

Dans cette question, vous devrez enrichir les formulaires d'édition suivants, en permettant la saisie et la sauvegarde des différentes relations possibles :

- Personne :
 - Service via une liste déroulante
 - Rôles via une liste de cases à cocher
- Rôles :
 - Personnes via une liste de cases à cocher

Il n'est pas demandé de pouvoir créer un rôle ou un service depuis le formulaire des personnes, ou réciproquement, de créer une personne depuis le formulaire des rôles. On considère que lorsque par exemple on affecte un rôle à une personne, le rôle existe déjà.

Question bonus (2 pts)

Si la question 2 a été réalisée en implémentant le pattern du repository, 2 points de bonus sont attribués.