



Module 1 Développer l'interface d'une application informatique

Composants accès données	Séance	S01	Activité	A-004B
--------------------------	--------	-----	----------	--------

Vous allez poursuivre dans cet atelier comment séparer les différentes couches de nos composants afin de faciliter la maintenance et la réutilisabilité du code.

Ce sera aussi l'occasion de découvrir de nouveaux patrons de conception sollicités couramment dans l'industrie du logiciel dans l'objectif de produire un code DRY, robuste et performant.

1 Gestion des prêts d'une bibliothèque

Vous devez mettre en œuvre les mécanismes d'enregistrement d'un emprunt d'un exemplaire en respectant l'architecture de composants définie et implémentée dans la maquette de l'application **BibliothequeNCouchesSQL**.

Créez la base de données **Bibliotheque** sur votre serveur.

Exécutez le script de création des tables et des données du jeu de test disponible en ligne.

Créez les procédures stockées relatives aux opérations **CRUD** à l'aide de l'outil de génération mis à votre disposition et disponible lui aussi en ligne.

Installez la maquette de l'application **BibliothequeNCouchesSQL** et testez celle-ci.

Etudiez les différents composants de l'application afin de vous approprier l'existant.

Complétez la maquette pour qu'elle puisse prendre en charge l'enregistrement d'un emprunt d'un exemplaire de livre par un adhérent.

Il vous faut créer les composants d'accès aux données nécessaires à l'implémentation de cette fonctionnalité.

Complétez les objets métiers avec les méthodes et contrôles de validité nécessaires en tenant compte des principes suivants :

- Un adhérent ne peut avoir plus de 5 exemplaires en emprunt (prêt en cours = absence de date de retour)
- Il n'est pas possible d'emprunter deux fois le même exemplaire à la même date. La liste des emprunts comporterait alors des doublons. Pour respecter les canons de l'objet, appuyez-vous sur la détermination de l'égalité selon les règles suivantes :
 - Deux emprunts sont identiques si Identifiant Adhérent, Identifiant Exemplaire et Date Courte Emprunt (sans Time) sont identiques. Pour la formule de hachage, prenez en considération les 3 éléments.

Les erreurs lèvent des exceptions qui devront être gérées au niveau de l'application cliente.

La méthode `EmprunterExemplaire` peut être implémentée au niveau de l'objet `Adherent`.

La liste des emprunts d'un adhérent est figurée par la propriété de navigation `Prets`.

Essayez d'acquérir de bonnes pratiques en externalisant les libellés de vos messages. Voir page suivante.

2 Externaliser les libellés

Définir les libellés des messages dans des fichiers de ressources plutôt que sous forme de littéraux constants dans nos programmes.

Une bonne préparation à une future internalisation de votre application ...

Commentaires sur les fichiers de ressources :

Une ressource est une donnée non exécutable qui sera déployée avec l'application. Les fichiers de ressources peuvent contenir des données de type chaîne, image ou l'état sérialisé d'un objet. Stocker les données dans des fichiers de ressources permet de modifier celles-ci sans recompiler l'application.

Il est important d'externaliser les données de l'application notamment pour pouvoir régionaliser cette dernière. Un fichier de ressources .resx se compose d'entrées XML, qui spécifient des objets et des chaînes dans des balises XML. Le .NET Framework fournit une prise en charge complète pour la création et la localisation des ressources.

Ensuite, au-delà de la régionalisation, cela permet à une tierce personne de définir les messages et libellés de l'application. Pas nécessairement inutile D'une manière générale, nous devons éviter d'encoder des constantes textuelles dans nos applications.

Vous pouvez modifier le modificateur d'accès d'internal à public pour permettre un accès plus large aux ressources.