

Diagramme de Classe

Notre modèle est divisé en deux parties distinctes, une bibliothèque « Données », qui contient toutes les classes représentant les différents éléments stockés dans notre base de données, et une bibliothèque « Gestionnaires », qui contient à la fois le Manager principal de l'application, qui gère et stocke la collection des données, mais aussi toutes les classes d'utilitaires, comme la recherche.

L'élément principal de ce diagramme est le dictionnaire Médiathèque. Il est stocké dans le Manager et crée le lien entre les Données et les Gestionnaires. Cette médiathèque a pour clés les objets « Ensemble Audio », basiquement la coquille d'une liste de musiques, cet objet servira concrètement à représenter une collection contenant des morceaux, des podcasts et des stations de radio. Un EnsembleAudio est lié à une énumération de genres par une propriété qui servira à le différencier dans l'utilitaire de recherche. Cet objet possède deux méthodes `ModifierImage` et `ModifierFavori` qui permettent d'attribuer une valeur aux attributs `CheminImage` et `Favori` directement dans la classe `EnsembleAudio` nous permettant ainsi d'utiliser des champs privés. Seule la propriété `Note` d'`EnsembleAudio` possède un setter public car pour plus d'ergonomie nous sommes obligés de faire un `Binding` sur un élément qui ne fonctionne qu'avec un mode `TwoWay`.

A cette clé Ensemble Audio est associé une valeur sous la forme d'une `LinkedList` d'objet `Piste`, un objet abstrait qui représente un élément qui peut prendre la forme soit d'un morceau, d'une station de radio ou d'un podcast. Chacun de ces objets qui héritent de `Piste` possède des propriétés et des méthodes différentes. On peut par conséquent stocker dans les `LinkedList` de `Piste` à la fois des podcasts, des morceaux ou des stations de radio, ce qui peut permettre à l'utilisateur de regrouper, dans des compilations thématiques par exemple, plusieurs types de médias (ex : un Ensemble Audio Jimi Hendrix pourrait très bien contenir les meilleurs morceaux de cet artiste ainsi que quelques podcasts documentaires le concernant). La classe `EnsembleAudio` et les classes qui héritent de `Piste` possèdent toutes des méthodes permettant de modifier leurs attributs, cela permet d'assurer une couche de contrôle sur l'accessibilité des éléments.

Un autre élément tout aussi important est notre objet `Manager`. Ce dernier, qui gère les opérations qui sont directement liées au dictionnaire principal (ajout ou suppression de couple clés/valeurs), contient une liste d'`EnsembleAudio` correspondants aux favoris (pouvant être modifiée la méthode `ModifierListeFavori`), une collection observable contenant les clés du dictionnaire `Mediatheque` (cette collection sert à l'affichage des différents Ensembles audio) mais aussi une liste des genres disponibles. La méthode `CreerEnsembleAudio` permet de générer un `EnsembleAudio` par défaut, en choisissant son titre, qui pourra ensuite être modifié par l'utilisateur.

Ce Manager a pour attributs d'autres sous Managers à qui il délègue des tâches : un Manager d'Ensemble Sélectionné, qui traite toutes les tâches lorsqu'une seule clé du dictionnaire est sélectionnée, comme l'ajout ou la suppression d'une de ses pistes. Il possède aussi un Manager de profil, qui stocke et gère toutes les informations relatives aux paramètres de l'application et du profil comme la couleur du thème ou la photo de profil. Enfin le Manager principal dispose d'un Manager lecteur, qui possède la référence de la clé du dictionnaire qui est en train d'être lue permettant au lecteur XAML d'accéder aux valeurs du dictionnaire (Piste) correspondant à des fichiers audios. Par défaut, lorsqu'on lit un album, la liste de lecture, elle aussi stockée dans ce Manager Lecteur, correspond à l'ensemble des pistes associées à l'ensemble audio en cours de lecture. Toutes les méthodes permettant de contrôler la lecture comme PlayPause, Next et Previous se trouvent directement dans le code-behind de MainWindow.

Enfin, un utilitaire de recherche est présent, utilisé à la fois par le Manager et le code-behind, il présente des méthodes assez polyvalentes, qui peuvent être utilisées dans des contextes variés. On peut par exemple effectuer une recherche par genre qui n'affichera que les ensembles audio correspondant à ce genre, ou encore une recherche par mot-clé qui affichera les ensembles dont le nom correspond, ou dont une des pistes a son titre ou son artiste qui correspond à la recherche. Enfin la méthode Recherche de l'utilitaire de recherche permet d'utiliser les 2 méthodes simultanément et donc d'effectuer une recherche par genre avec une recherche par mot-clé.