

Exercice 1 MVC.NET

Pour faire les exercices, vous devez au préalable avoir:

→ Créé un projet Web-Visual Studio 2013-Application Web ASP.NET MVC 4 avec l'option *Créer un répertoire pour la solution* cochée.

→ Ajouté le modèle Movie (clique droit sur Models>add>Class>Movie) :

Utiliser « prop tab tab » pour générer rapidement les propriétés

```
public int ID { get; set; }  
public string Title { get; set; }  
public string Genre { get; set; }
```

COMPILER (Ctrl-Shift-b) et non juste sauvegarder.

→ Ajouté un contrôleur nommé *Movie* (clique droit sur Controllers>add>Controller). Choisir l'option *Controller with empty read-write actions*, ne pas prendre celui avec Entity

→ Ajouté la référence sur le modèle *Movie* dans *MovieController*:

```
using [nomProjet].Models;
```

→ Créé la vue *Index*, pour se faire vous devez avoir:

1. Créé le répertoire *Movie* dans le répertoire *Views*.
2. Ajouté la vue *Index* (clique droit répertoire *Movie*, ajouter->View, Create) et avoir choisi les options suivantes :
 - Moteur de vue: Razor (CSHTML);
 - Créer une vue fortement typée cochée;
 - Classe de modèle: *Movie* (NOMPROJ.Models);
 - Modèle de vue de structure: *List*

→ Modifié dans la classe *MovieController*, la méthode ***Index*** et temporairement retourner une liste de *Movies* vide à la vue *Index*

```
return View(new List<Movie>());
```

Première partie du travail à effectuer :

ÉCRIRE LE CODE DEMANDÉ PAR LES QUESTIONS SUIVANTES:

- 1) Ajouter, dans le répertoire *Models*, une classe nommée *Repository*.
- 2) Implémenter, dans la classe *Repository*, une méthode nommée *GetAllMovies* qui retourne la liste de tous les films. Puisque que nous n'avons pas encore de base de données dans le projet, codez directement les créations des instances de films.

Voici le code à produire (seule question où le code est fourni!):

```
public List<Movie> GetAllMovies()
{
    Movie m1 = new Movie
    {
        ID = 1,
        Title = "Pulp Fiction",
        Genre = "Inclassable"
    };
    Movie m2 = new Movie
    {
        ID = 2,
        Title = "Lord of the rings",
        Genre = "Adventure"
    };
    List<Movie> list = new List<Movie>();
    list.Add(m1);
    list.Add(m2);
    return list;
}
```

- 3) Modifier, dans la classe *MovieController*, la méthode *Index* afin qu'elle puisse passer à la vue *List* la liste de films générée par la méthode *GetAllMovies* du *Repository*.
- 4) Ajouter un onglet dans la page principale du site qui donne accès à la Liste de Films.

Indice, vous devez modifier la vue *shared/_layout*:

- 5) Implémenter, dans la classe *Repository*, une méthode nommée *GetMovie* qui reçoit un id et retourne le film correspondant. Puisque que nous n'avons pas encore de base de données dans le projet, codez directement les valeurs des propriétés du film (tenez compte du id reçu en paramètre).
- 6) Implémenter dans la classe *MovieController*, la méthode **Details** qui reçoit un id et retourne la vue appropriée (créer la vue si nécessaire) en lui passant le film correspondant au id reçu en paramètre (utiliser la méthode *GetMovie* du *Repository*).
- 7) Implémenter, dans la classe *Repository*, une méthode nommée *UpdateMovie* qui reçoit un film et le met à jour dans la base de données. Puisque que nous n'avons pas encore de base de données dans le projet, nous allons laisser le corps de cette méthode vide.
- 8) Implémenter, dans la classe *MovieController*, la méthode **Edit** qui reçoit un id et retourne la vue appropriée (créer la vue si nécessaire) en lui passant le film correspondant au id reçu en paramètre (utiliser la méthode *GetMovie* du *Repository*).
- 9) Implémenter, dans la classe *MovieController*, la méthode *Edit* qui reçoit un film et simule la mise à jour de ce film dans la BD (utiliser la méthode *UpdateMovie* du *Repository*). Le cadre d'applications s'occupe lui-même d'instancier le film (à partir des données présentent dans la vue) et de le passer à la méthode *Edit*. Et c'est parfait ainsi!
- 10) Implémenter, dans la classe *Repository*, une méthode nommée *AddMovie* qui reçoit un film et l'ajoute dans la base de données. Puisque que nous n'avons pas encore de base de données dans le projet, nous allons laisser le corps de cette méthode vide.
- 11) Créer la vue *Create* qui permet de créer un Film. Le contrôleur *MovieController* doit-il être modifié afin de pouvoir appeler cette vue? A vous de vérifier votre compréhension du cadre d'applications MVC.
- 12) Implémenter, dans la classe *MovieController*, la méthode **Create** qui reçoit un film et qui simule la création de ce film dans la BD (utiliser la méthode *AddMovie* du *Repository*).

- 13) Implémenter, dans la classe *Repository*, une méthode nommée *DeleteMovie* qui reçoit un film et supprime le film correspondant de la base de données. Puisque que nous n'avons pas encore de base de données dans le projet, nous allons laisser le corps de cette méthode vide.
- 14) Créer la vue *Delete* qui permet de supprimer un Film.
- 15) Ajouter le code nécessaire dans la classe *MovieController* afin que la vue *Delete* précédemment créée puisse s'afficher correctement.
- 16) Implémenter, dans la classe *MovieController*, la méthode **Delete** qui reçoit un id et simule la suppression du film correspondant au id dans la BD (utiliser la méthode *DeleteMovie* du *Repository*).
- 17) Faites un ré-usinage de la classe *MovieController* afin de ne plus avoir à instancier la classe *Repository* avant chaque utilisation.