# **TP2**

## Quelques pistes et aide

### Importance de communiquer dans l’équipe

### Les formatifs et le Tp1 ont beaucoup d’éléments que vous pouvez récupérer.

### Ordre de développement (difficulté progressive)

* 1. Page d’entrée juste l’affichage (formatif 2B jumbotron et le formulaire s’enregistrer)
  2. Modèle pour le média (Tp1 classe, enregistrer et lire en xml)

Sérialiser la liste de média pour créer un fichier XML

* 1. Page des nouveautés

Faire le gabarit des pages (Formatif 3 barres de navigations, sections, layout) aucune action n’est programmée, juste prévoir l’affichage des nouveautés sans action sur sélection et sans association à un utilisateur

Programmer la page des nouveautés (Formatif 2b – caroussel construit avec le modèle, boucle Razor)

Programmer l’ajustement avec de nouvelles nouveautés

* 1. Page de l’offre

Les anciennes nouveautés vont dans cette page

La page des nouveautés montre une nouvelle offre

* 1. Login

Avoir un objet utilisateurs et créer 2 utilisateurs dans un fichier xml et un utilisateur admin

Réfléchir pour sérialiser désérialiser en XML avec toutes sortes d’objets

Réfléchir pour un utilisateur courant

* 1. Ma Liste

Réfléchir où conserver les préférences de l’utilisateur

* 1. Traitement sur les médias (jouer, appréciation, description etc.)

### Des instructions utiles

* chemin des images particularités

Exemple : @{var monImage = Model[0].NomImage; }

<img src="~/Content/images/@monImage"

* chemin des fichiers xml

string monFichier = HttpContext.Current.Server.MapPath(@"~/App\_Data/" + nomFichier);

* construction des listes et des données persistantes pour la durée de l’application

Mettre les objets persistents (liste de films, utilisateurs, etc.) dans le fichier global.asax

public static List<Film> films { get; set; }

public static List<Utilisateur> utilisateurs { get; set; }

Peupler les listes dans cette classe (MvcApplication) dans des méthodes privées.

**Dans le contrôleur, récupérez les données ainsi, par exemple :**

var utilisateurs = MvcApplication.utilisateurs as List<Utilisateur>;

return View(MvcApplication.films);

TempData["Utilisateur"] = MvcApplication.utilisateurActif as Utilisateur;

**Dans une vue, par exemple :**

var util = TempData["Utilisateur"] as Utilisateur;

ou avec @**Model** si on a passé le modèle en paramètre