

*Formation DWWM / 2020-2021*

**TP Symfony** /

1. Menus :

Pour démarrer la mise en place de l’application de notre application, nous allons proposer différemment des menus pour notre interface.   
Ce menus utilisera le CSS/ JS & JQueries en regles avec la politique de l’agence

* 1. **User Menu** *(base User :Menu)*
  + Accueil
  + A propos
  + Les Ventes
    - Terrains,
    - Appartements,
    - Maisons,
    - Hôtels
  + Les locations
    - Terrains,
    - Appartements,
    - Maisons,
    - Hôtels
  + Test
  + Nous Contacter
  + Connexion
  + Administration

1. **Admin Menu** *(base Admin:Menu)  
   Accueil / Système / Users / Contenu / Extensions / Help / Logout*
   * Administration (arborescence)
     + Accueil
     + Système *( Page Config du Système)*
     + Utilisateurs *(Gestion des utilisateurs voir Entité Utilisateurs au N°4)*
     + Contenu
       - Catégorie (Gestion des Catégories Categorie au n°4)
       - Location (Gestion des locations)
       - Achats & Ventes (Gestion des Achats & Vente)
       - Articles à la UNE (Gestion des Articles à la Une)
       - Images (Gestion des Images)
     + Extensions (activation & désactivation de Modules/Plugin)
     + Logout (sortir de l’admin)
2. **Footer**
   1. Faire une proposition de FOOTER selon la méthode de votre choix
   2. Diviser cette region en 3 differents blocs
3. **Entités / Relations /**
   1. Mettre en place des tables qui seront en relation avec les contenus des Biens en Location et en vente,
   2. Voici les tables et leurs structures:
      1. **Location**
   * Date;
   * Titre (ou Denomination);
   * **Categorie**
   * Image/photo ;
   * Description ;
   * Surface;
   * Type-maison;
   * Nombre de Chambres;
   * Nombre d”étages;
   * Prix en Euros;
   * Adresse
     + Code postal;
     + Ville;
     + Pays
   * Accessibility ;
   * **Statuts**(actif/occupé/desactivé) ;
   * **A la UNE** (oui/Non)
4. **Categorie**
   * **Titre,**
   * **Resumé,**
5. **n**
   * Noms,
   * Prénoms
   * Photo
   * Date Naissance
   * Login
   * Password,
   * Adresse,
   * Email,
   * Rôle
     + Locataires,
     + Propriétaires,
     + Gestionnaires,
     + Administrateurs
6. **Formulaire :**
   1. Dans votre application personnelle, Mettre en place le formulaire de Gestion de chaque Entité (Location / Catégorie / Utilisateurs),
   2. Rendre obligatoire tous les champs des formulaires,
   3. Les Contraintes prennent en compte les types de chaque propriété,
   4. Afficher les biens saisis dans (Vente / Location)
7. **Formulaire** (suite)
   1. Donner à l’administrateur la possibilité d’avoir la Liste des Entités (Location / Vente / Ustilisateurs / Catégories), avec la possibilité d’éditer, de supprimer ou d’ajouter de Nouveau Bien.
   2. Profiter également pour rendre (Editable/Supprimable) tout Bien en Affichage. Cette option s’exécutera en Rôle «Admin »
8. **Formulaire** (suite)
   1. Mettre en place une Entity Catégorie, avec les propriétés suivantes :
      * + Titre (Label)
        + Résumé
      1. Mettre en place une association avec l’entité Location. Le valeur de la clé ne dépendra que de vous.
      2. Mettre à jour le formulaire d’enregistrement & de MAJ pour qu’ils prennent en compte cette association
   2. Forcer les formulaires à utiliser les valeurs prédéfinies pour les champs :
      1. Type (F1-F6)
      2. Nombre de chambres
      3. Nombre d’étages,
      4. Accessibilité
   3. Créer une Entité User(utilisateur) dont les propriétés sont les suivantes :

Noms / Prénoms / Date Naissance / Login / Password / Adresse / Email / Rôle

* + 1. Rendre actif la MAJ d’un utilisateur,
    2. Lier cet utilisateur à un Bien (nous travaillerons cette partie à la rentrée)

1. **Gestions de la UNE**
   1. **Activation/ DEsactivation,**
   * Afficher à la UNE que les Biens selections,
   * Les afficher dans l’ordre « DESC »
   * Rendre invisibles pour tous les utilisateurs tous les Biens déjà en Location,
   1. **Animation Caroussel** (Sélection des Articles +)
      * + - L’équipe (Team)
          - Dernier Appartements (1)
          - Dernière Maison (1)
2. **CSS / JS** (travail d’équipe)
   1. Mettre en place, selon la chartre graphique de la classe, un modele (css/Js) qui sera utilisé tout au long du projet.
   2. Définir les tailles et les couleurs des balises ainsi que leurs comportements.
3. **Repository**
   1. **Filtre de de Biens (Enrichir le Filtre de la Page 1)**
      * Mettre en place 1 filtre pour la gestion des Bien par Sous-Catégorie   
         (Appartements/Maisons/Terrains/Hotels)
      * Ce Filtre par SousCatégorie prendra également en compte les propriétés suivantes
        + PrixMaxi –PrixMini
        + SurFaceMaxi - SurfaceMini
        + Types (F1/F2/F3/F4/F5)
        + Localité
        + Accessibilité
      * Utilisez ce Model de filtre sur les pages qui gèrent les sous-menu (Maison & Appartement)

Voici le modèle final du Filtre

Filtrer

**Categorie**

* Appartements
* Maisons
* Terrains
* Hotels

**Critères**

* PrixMaxi – PrixMini
* SurFaceMaxi – SurfaceMini
* Types (F1/F2/F3/F4/F5)
* Localité
* Accessibilité

1. **ASSETIC** (CSS / JS / IMAGES)  
   1. Bootstrap (JS/CSS)
      * Télécharger et remplacer tous les CSS et JS de BootStrap utilisés dans la base (base.htm.twig) du Socle
      * Placer ces fichiers dans l’emplacement de choix,
      * Configurer votre WEBPACK pour une prise en main de ces fichiers
      * Rendre Fonctionnel votre socle avec ces fichiers de Bootstrap
   2. **NPM / YARN**
      * Installez sur votre socle Npm ou Yarn selon votre convenance et rendez
      * Rendre votre Socle fonctionnel (en Localhost)
   3. **CSS Perso**

Proposer 1 fichier CSS (app2.css) qui animera votre socle avec les caractéristiques suivantes :

* + - Header / Footer
      * une même couleur de fond
    - Body
      * couleur de fond = blanche,
      * Police = verdana
      * Taille = normale
    - h1
      * Alignement centré
      * Coleur = bleu
    - h2
      * couleur = rouge
    - Paragraphe / Enumération / et le Tableaux :
      * Couleur de texte = noir