 Examen formatif

0% de la note finale

Durée : 2 heures et 15 minutes

**Notions évaluées :**

* Création d’applications Web Asp Net Core
* Requêtes Linq et Injection de dépendances
* Sections (Vues), routage par attribut et passage de données
* Helpers, Assistants et validation

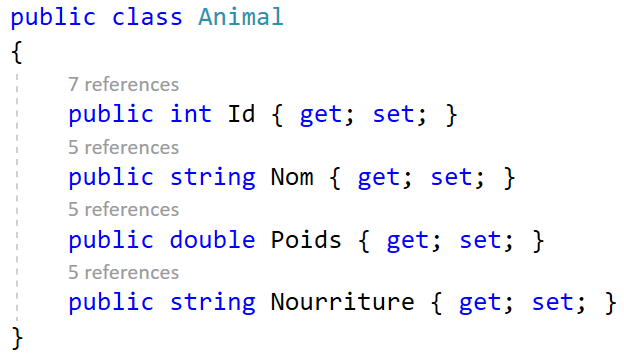
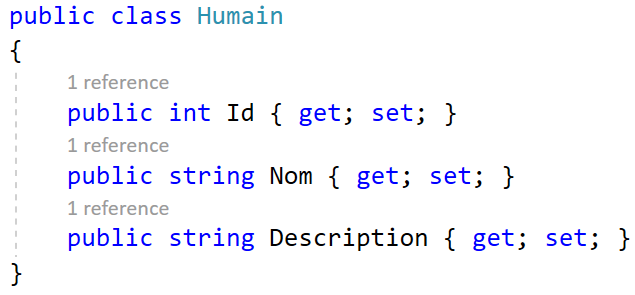
**Département d'informatique**

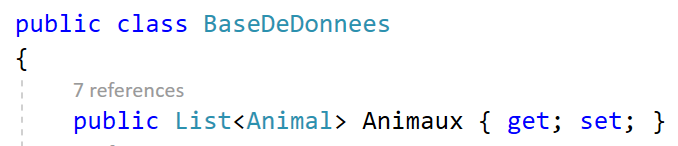
* Vues partielles, variables de session et redirections
* Passage de données du contrôleur à la vue (Model, ViewData, ViewBag et ViewModel)
* Disposition de base avec Bootstrap
* Règles CSS simples

Vous pouvez utiliser toute la documentation que vous souhaitez, tant que l’examen est fait individuellement.

Instructions de remise

* **Sauvegardez fréquemment** votre projet Asp Net Core.
* Lors de l’examen sommatif final, une fois terminé, compressez le dossier racine de l’examen et tout son contenu et remettez-le sur la plateforme utilisée par votre enseignant.
* Prenez connaissance des classes BaseDeDonnees, **Animal** et **Humain** pour mieux comprendre les questions de l’examen :

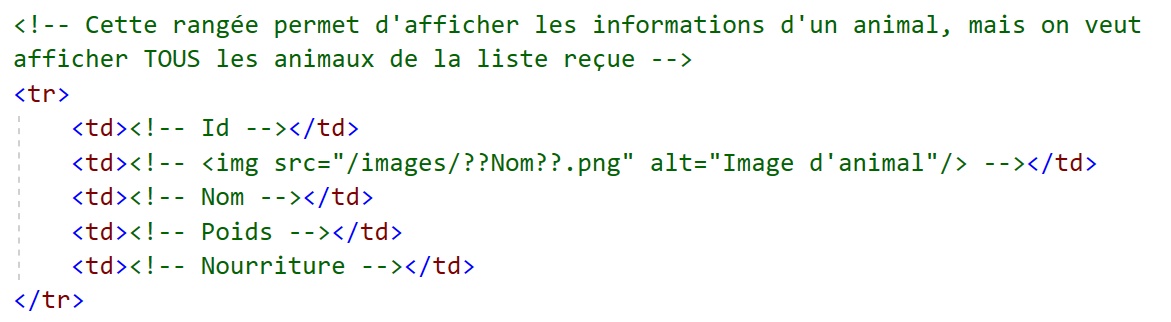


Une vue pour l’enclos de Steve



Pour aider Steve, il va falloir compléter la vue **/Ferme/Enclos** qu’il a commencée. Ouvrez le projet Asp Net Core dans le dossier de l’examen avec Visual Studio. Notez que pour le moment, l’application fournie compile sans problème.

1. Dans la vue **/Ferme/Enclos**, on a besoin d’un objet de type **List<Animal>** qui doit être envoyé par le contrôleur **FermeController**. (Action **/Ferme/Enclos**)
   1. Servez-vous de la liste d’animaux de la **BaseDeDonnees** pour ceci.
2. Dans la vue **/Ferme/Enclos**, on souhaite afficher quelques infos :
   1. On veut afficher le **nom** et la **description** de l’**humain**. (L’humain est créé dans la vue et est stocké dans la variable nommée « **steve**»)
   2. On veut afficher 5 informations spécifiques qui portent sur la **liste d’animaux** un peu plus loin dans la vue :
      * Nombre d’animaux dans la liste.
      * Poids moyen des animaux dans la liste.
      * L’enclos est-il bondé ?
        + Afficher « Oui » si la liste d’animaux reçue contient au moins 10 animaux.
        + Afficher « Non » sinon.
      * Poids le plus élevé (maximal) des animaux dans la liste.
      * Afficher les données dans le ViewBag à la clé "Enclos"
   3. On veut afficher **tous** les **animaux** de la **liste** dans un **tableau**. (Un animal par rangée) Le code nécessaire pour afficher une rangée est déjà présent. Essayez d’éviter de dupliquer du code grâce à une boucle.
      * Pour l’**image**, l’attribut **src** doit contenir **/images/**, le **nom** de l’animal et **.png**. (Exemple : **src="/images/Pig.png"**)



1. Enfin, transformez chaque rangée en **vue partielle**.
   1. Ça veut dire que le code ci-dessus (<tr>...</tr>), une fois complété, devra être encapsulé dans une vue partielle. (Et adapté pour pouvoir fonctionner dans cette vue partielle)

Une vue pour la maison d’Alex

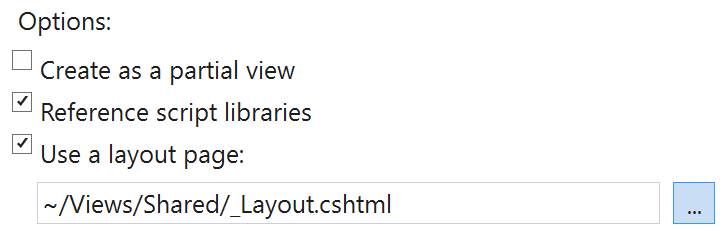
Pour aider Alex, il va falloir légèrement modifier la vue **/Ferme/Maison**. Nous allons modifier légèrement la disposition à l’aide de classes Bootstrap et nous allons ajouter quelques règles CSS.

1. Présentement, il y a deux éléments **<div>** dans la vue **Maison.cshtml**. Utilisez des classes **Bootstrap** (Et ajoutez des éléments HTML, s’il le faut) pour que ces deux éléments **<div>** soient deux **colonnes** côte-à-côte.
   1. La colonne qui contient les souhaits d’Alex devrait prendre **1/4** de la rangée, et l’image de sa maison devrait prendre **3/4** dans la rangée.
   2. Vous n’avez pas à créer un « **container** » ! Le **Layout** s’en occupe déjà.
2. Ajoutez également les règles CSS nécessaires dans la **@section Style** afin de... :
   1. Retirer le **padding à gauche** de la liste des souhaits.
   2. Mettre le mot « zombies » en vert. (Vous devrez ajouter un élément HTML pour ça)
   3. Mettre la couleur de fond « **rgba(0,0,0,0.8)** » pour la colonne qui contient les souhaits d’Alex.

Une vue pour créer des animaux

Finalement, on veut créer une vue **/Ferme/Create** à l’aide d’un assistant. Cette vue nous permettra de créer des objets de type « **Animal**», que nous pourrons ajouter à la **base de données**.

1. Voici les options à utiliser dans l’assistant : (Vous devrez savoir quoi mettre pour les 3 premiers champs)



1. Retirez le champ pour l’Id. Il y a déjà du code dans l’action [HttpPost] **/Ferme/Create** qui vous permet de choisir un Id approprié.
2. Ajoutez le nouvel Animal reçu par l’action **/Ferme/Create** à la base de données.
3. L’action **/Ferme/Create** doit rediriger vers l’action **/Ferme/Enclos** si l’ajout a fonctionné.
4. Pas besoin de modifier le lien « **Back to List** » dans le bas de la page.
5. Il faut ajouter les étiquettes et contraintes (Grâce à des **attributs de modèle** / **annotations**) pour les champs **Nom**, **Poids** et **Nourriture** :
   1. D’abord, mettre des **étiquettes** personnalisées:
      * **Nom** : « Le nom de l’animal »
      * **Poids** : « Le poids de l’animal »
      * **Nourriture** : « La nourriture de l’animal »
   2. Ensuite, rendre **obligatoires** les propriétés **Nom**, **Poids** et **Nourriture**.
   3. Puis, ajouter les contraintes suivantes
      * **Poids**: La valeur numérique doit être située entre 0.5 et 2000
      * **Nourriture** : La taille maximale de la chaîne de caractères est 30
   4. Enfin, personnaliser les **messages d’erreur** :
      * **Nom**  (Aspect obligatoire) : « Entrez un nom. »
      * **Poids** (Aspect obligatoire) : « Entrez une poids. »
      * **Poids** (Aspect valeur) : « Entre 0.5 et 2000. »
      * **Nourriture** (Aspect obligatoire) : « Indiquer la nourriture. »
      * **Nourriture** (Aspect taille) : « 30 caractères maximum. »
6. Finalement, modifier l’action Create (POST) afin de valider l’état du modèle avant son intégration dans la base de données. Si le modèle n’est pas valide, réafficher la vue.

Si vous voulez tester votre vue **Create.cshtml**, vous pouvez tenter de créer un animal nommé « Horse » avec un poids de votre choix et de la nourriture de votre choix. Un cheval devrait apparaître dans la vue **/Ferme/Enclos** dans le tableau d’animaux