

Consignes Séquence 4 : Contrôleurs

Création des premiers jeux de tests et de l'entité Car

Exercice 3

Vous allez créer un **premier jeu de données test**.

Dans le contrôleur « **CarController** » créé précédemment, **initialiser un tableau associatif** de constantes tel que :

- La marque de la voiture soit Ferrari
- Le modèle de la voiture soit 488 GTB
- L'année de commercialisation soit 2015
- Le moteur soit V8 bi-turbo
- La couleur soit rouge

Une fois le tableau initialisé, envoyez les données de celui-ci à la vue **Car/index.html.twig** grâce à la méthode `index()`.

Finalement, sur le **template Car/index.html.twig**, afficher les données de la voiture.

Exercice 4

Vous allez pousser un peu plus loin la création des jeux de données test.

Créez une entité « Car », dans le dossier **src/Entity**. Elle devra être composée des champs suivants :

- **brand**, string, 255 caractères, non nul
- **model**, string, 255 caractères, non nul
- **year**, integer, non nul
- **engine**, string, 255 caractères, non nul
- **color**, string, 255 caractères, non nul

Dans cette entité, on devra trouver :

- Les **attributs** correspondant aux caractéristiques de la voiture précédemment créée (marque, modèle, année de commercialisation, moteur et couleur). Ils ne doivent être accessibles que dans ce fichier **uniquement**.
- Les **méthodes GET/SET** de ces différents attributs. Ils devront être « **fluent** » pour faciliter les manipulations futures.

Ensuite, de retour dans le **CarController**, créer la méthode magique **__construct()**.

Dans ce constructeur, vous devrez **instancier la classe Car** nouvellement créée et **initialiser les getters et setters** avec les valeurs de votre choix (attention tout de même à respecter la thématique).

Une fois le constructeur réalisé, passer les données nouvellement générées à la vue et les afficher sur le **template Car/index.html.twig**.