Exercices de la formation sur Angular

# Exercice 1

Créer deux composants login et home

Component Login :

* Contient un formulaire (login et mot de passe avec bouton enregistrer)
* Doit avoir un plus du formulaire le titre de l'application en haut à gauche
* Le clic sur enregistrer nous ramène vers la page home

# Exercice 2

Créer un module components sans routage avec les composants suivants :

* success message component: qui reçoit un message en input et l'affiche dans une alarte
* de même créer un danger message component

# Exercice 3

Le home component est constitué :

* navbar component (à mettre dans le module components)
* sideBar component (à mettre dans le module components) : va contenir les menus de l’application
* et router outlet qui héberge les sous routes de home

# Exercice 4

Mettre des données dans le service data à l'initialisation de l'application

* mettre des utilisateurs (login et mot de passe) dans le service data
* isoler les données dans un fichier json (db.json), lire le fichier et mettre son contenu dans le data service au démarrage de l'application

# Exercice 5

* Créer un service data pour stocker les données (par exemple les infos de l’utilisateur connecté)
* Utiliser session storage pour stocker les données de l’utilisateur connecté après le chargement de la page

# Exercice 6

Authentification : utiliser les données qui sont dans le fichier json

* créer auth service avec 2 méthodes logIn(user) et logOut()
* utiliser ce service lors de l'authentification et lorsqu'on se déconnecte
* si l'utilisateur n'est pas présent dans la base (db.json) afficher un message d'erreur et rester dans la page login sinon routage vers le home et enregistrer l'utilisateur courant (l'utilisateur avec lequel on est connecté)

# Exercice 7

Ajouter un guard sur la route home

si l'utilisateur courant n'existe pas router vers login (si quelqu'un met la route du home avant de se connecter il sera rediriger vers login)

# Exercice 8

Créer les services root avec l'annotation root

* util service pour les fonctions utilitaires par exemple ordonner une liste par une clé donné
* refs pour les référentiels: les types de voitures

# Exercice 9

créer un dossier models qui héberge les models (par exemple IUser : interface pour l’utilisateur qui va contenir : id, nom, genre ….)

# Exercice 10

Créer un dossier enum dans laquel on met les enumérations globales

par exemple :

export enum CouleurVoiture { WHITRE, BLACK, YELLOW, GREEN }

export enum EtatVoiture { Nouvelle, Moyenne, Ancienne }

# Exercice 11

Créer un dossier pages dans lequel on mettra les modules avec routage: les pages de l'application

# Exercice 12

Dans le dossier pages créer un module avec routage : refs

CRUD des référentiels : créer des composants réutilisables pour le référentiels (list, frm)

Chaque réf contient (code, lib)

Les réfs sont marque, année modèle, couleur, type, etat, profils (admin, contrôleur, DG)

Point d’entrée : le menu « gestion des référentiels».