

Examen Pratique (TP) - Développement Front-end avec Angular

Durée : 1h30 Type : Pratique sur PC Note : 20 points

Contexte

Vous êtes chargé de développer une application de gestion de bibliothèque simple en utilisant le framework Angular. Cette application doit permettre de gérer une collection de livres, en offrant des fonctionnalités de consultation, d'ajout, de suppression et de recherche.

Pré-requis (À réaliser avant l'examen)

Pour gagner du temps, vous devez avoir préparé un nouveau projet Angular à la maison.

- Création du Projet :** Créez un nouveau projet Angular avec la commande `ng new Projet-NomPrenom` (Exemple : `Projet-GuerboujSonia`).
- Intégration de Bootstrap :** Installez et configurez Bootstrap dans votre projet.
- Composant Menu :** Créez un composant de menu nommé `votreNom-menu` (Exemple : `guerbouj-menu`) et intégrez-le dans le composant racine (`AppComponent`). Ce menu doit être stylisé (en utilisant Bootstrap) et contenir les liens de navigation pour les futures pages.

Partie 1 : Modèle et Service (5 points)

L'application gère des livres. Chaque livre est défini par les informations suivantes :

Champ	Type	Description
<code>id</code>	<code>number</code>	Identifiant unique du livre
<code>titre</code>	<code>string</code>	Titre du livre
<code>auteur</code>	<code>string</code>	Nom de l'auteur
<code>anneePublication</code>	<code>number</code>	Année de publication
<code>genre</code>	<code>string</code>	Genre du livre (Ex: Science-Fiction, Roman, etc.)

disponible

boolean

Indique si le livre est disponible ou non

Tâches

- Création du Modèle (1 pt) :** Créez une classe modèle nommée `Livre` dans un fichier `livre.model.ts` qui respecte la structure ci-dessus.
- Création du Service (2 pts) :** Créez un service injectable nommé `LivreService`. Ce service doit contenir :
 - Une propriété privée `livres: Livre[]` initialisée avec au moins 5 objets `Livre` pour simuler des données.
 - Une méthode publique `getLivres(): Livre[]` qui retourne la liste complète des livres.
- Méthode d'Ajout (2 pts) :** Ajoutez une méthode `ajouterLivre(nouveauLivre: Livre): void` au service pour ajouter un nouveau livre à la liste.

Partie 2 : Composants et Affichage (7 points)

Tâches

- Composant Liste (3 pts) :**
 - Créez un composant nommé `LivreListComponent`.
 - Injectez le `LivreService` dans ce composant.
 - Affichez la liste des livres dans un tableau HTML en utilisant la directive `@for`.
 - Utilisez la directive `@if` pour afficher un message si la liste des livres est vide.
 - Utilisez la directive `ngClass` pour mettre en évidence (par exemple, en vert) les lignes des livres dont l'année de publication est supérieure à 2020.
 - Utilisez la directive `ngStyle` pour colorer en rouge le texte du titre si le livre n'est pas disponible .
- Composant Ajout (4 pts) :**
 - Créez un composant nommé `LivreAjoutComponent`.
 - Créez un formulaire pour saisir les informations d'un nouveau livre (Titre, Auteur, Année de publication, Genre, Disponible).
 - Utilisez la directive `ngModel` pour la liaison bidirectionnelle des données du formulaire.

- Lors de la soumission du formulaire, appelez la méthode `ajouterLivre` du service et réinitialisez le formulaire.

Partie 3 : Routage et Directives Avancées (8 points)

Tâches

1. Configuration du Routage (3 pts) :

- Configurez le routage dans `app.routes.ts` pour les chemins suivants :
 - `/liste` : Affiche le `LivreListComponent`.
 - `/ajouter` : Affiche le `LivreAjoutComponent`.
 - `/` (chemin par défaut) : Redirige vers `/liste`.
- Utilisez les liens de navigation dans votre composant `votreNom-menu` pour permettre la navigation entre ces deux vues.

2. Composant Recherche et Suppression (5 pts) :

- Créez un composant nommé `LivreGestionComponent`.
- Dans ce composant, ajoutez un champ de saisie pour la recherche par **Genre** et une méthode pour filtrer la liste des livres.
- Affichez la liste filtrée.
- Ajoutez un bouton "Supprimer" à côté de chaque livre. Implémentez la logique de suppression dans le service etappelez-la depuis ce composant.
- Utilisez la directive `@switch` dans l'affichage de la liste pour afficher un badge de couleur différente en fonction du `genre` du livre (par exemple, bleu pour "Science-Fiction", vert pour "Roman", jaune pour "Autre").

Fin du Sujet