

Semestre : 1 ☒ 2 ☐

Session : Principale ☒ Rattrapage ☐

Module : Application coté client

Enseignant(s) : Ameni ROMMENE, Radhouene MASSOUDI

Classes : 4TWIN

Documents autorisés : OUI ☒ NON ☐ Nombre de pages : 02

Calculatrice autorisée : OUI ☐ NON ☒ Internet autorisée : OUI ☐

NON

Date : 15/01/2020

Heure: 09h

Durée:1h30

NB :

- Il faut utiliser le projet envoyé sinon vous aurez 0.
- Pour lancer votre projet tapez la commande : **npm start** (car il y a la configuration de proxy dans le projet envoyé)
- Pour lancer json-server, tapez la commande : **json-server --watch db.json**
- Les urls de json-server suite à la configuration de proxy sont de la forme /api/entité
- Toutes les méthodes du service http doivent être développées dans le service ProductService.

Travail demandé :

- 1- Au niveau du composant AppComponent, ajoutez 3 liens **add**, **home** et **Shopping Cart** qui pointent respectivement sur les composants **AddProductComponent**, **HomeComponent** et **showShoppingComponent**. (3 pts)
- 2- Au niveau du composant **AddProductComponent**, ajoutez un formulaire d'ajout d'un produit comme indiqué dans la **Figure 1**

Name	<input type="text"/>
Price	<input type="text"/>
Description	<input type="text"/>
<input type="button" value="Add Product"/>	

Figure 1

- 3- Affichez un message adéquat pour chaque contrainte non respectée au niveau du formulaire sachant que :
 - 🚫 Name **obligatoire (0.75 pt)**
 - 🚫 Price ne contient **que des caractères numériques** (utilisez un **pattern**) (1 pt)
 - 🚫 Description contient **au moins 10 caractères**. (0.75 pt)
 - 🚫 Le bouton « Add Product » est activé si toutes les conditions précédentes sont respectées. (1 pt)

- 4- Créez un service **ProductService**, dans lequel vous définissez une fonction appelée **addProduct()** qui permet d'ajouter un produit au niveau du fichier **db.json** et plus précisément dans le tableau **products**.
- 5- En cliquant sur le bouton « **addProduct** » :
 - Le produit est enregistré au niveau du fichier **db.json** en appelant la méthode **addProduct()** déjà créée. (2.5 pts)
 - Une alerte **"Product Succesfully added"** s'affiche. (1 pt)
- 6- Au niveau du composant **HomeComponent** affichez la liste des produits contenus dans le fichier **db.json**, tableau **products** comme indiqué dans la **Figure 2**. Les produits s'affichent une fois le composant est chargé (2 pts)



Figure 2

- 7- En cliquant sur le bouton « Buy » de la Figure 2, le produit **concerné** est ajouté dans le fichier **db.json** et plus précisément dans le tableau **cart**. (2 pts)
- 8- Au niveau du composant **showShoppingComponent** affichez la liste des produits que vous avez achetés et qui sont contenus dans le tableau **cart**. (2 pts)
- 9- Afin d'améliorer l'action d'achat d'un produit, ajoutez la condition suivante en cliquant sur le bouton « buy » : Si un produit existe déjà dans le tableau **cart**, alors affichez une alerte « Product already exist » et le produit ne s'ajoute pas. Sinon le produit s'ajoute dans **cart**. (2 pt)
- 10- En cliquant sur le bouton « Details » de la Figure 3, une **redirection** se fait vers un composant appelé **DetailsComponent** (voir annexe), que vous devez créer, et qui affichera la description du produit **concerné** comme indiqué dans la **Figure 3**. (2 pts)

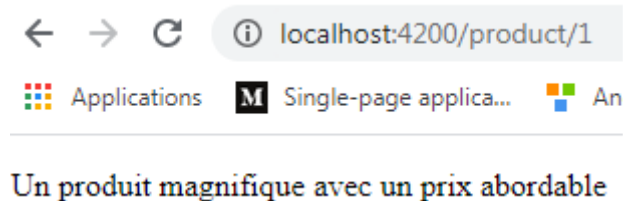


Figure 3

Annexes :

La méthode **navigate** du **serviceRouter** importé depuis **@angular/router** permet de faire une redirection vers un path déjà défini.

- Exemple sans paramètres à envoyer : **this.router.navigate(['path'])**
- Exemple avec un paramètre appelé « param » à envoyer : **this.router.navigate(['path/'+param])**